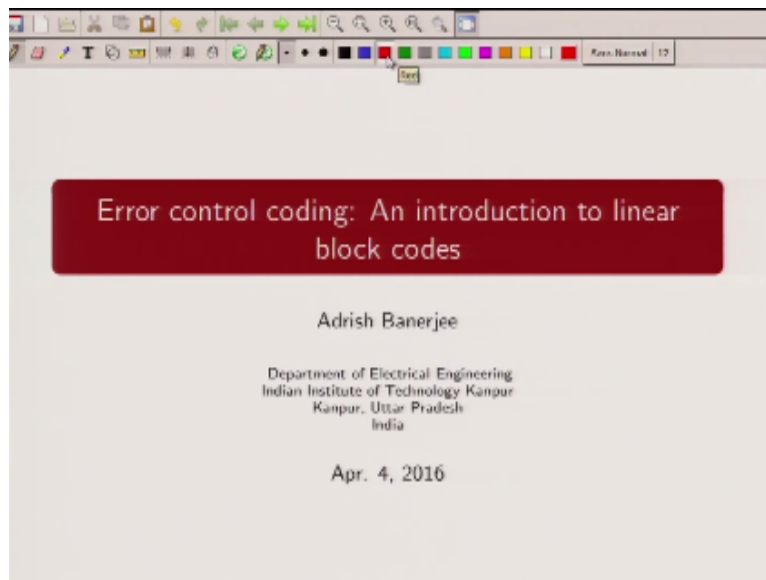


**Indian Institute of Technology Kanpur**  
**National Programme on Technology Enhanced Learning (NPTEL)**  
**Course Title**  
**Error Control Coding: An Introduction to Linear Block Codes**

**Lecture-9C**  
**Decoding of low density parity check codes-II: Belief**  
**Propagation Algorithm**

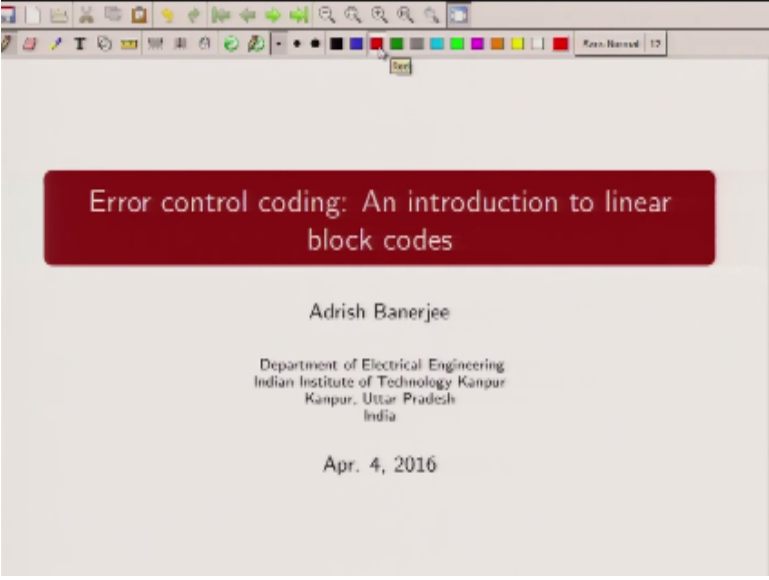
**by**  
**Prof. Adrish Banerjee**  
**Department of Electrical Engineering, IIT Kanpur**

(Refer Slide Time: 00:15)



Welcome to the course on error control coding, an introduction to linear block codes.

(Refer Slide Time: 00:21)



The image is a screenshot of a presentation slide. At the top, there is a toolbar with various icons for navigation and editing. The slide has a light gray background. A dark red rectangular box is centered on the slide, containing the title "Error control coding: An introduction to linear block codes" in white text. Below the title, the author's name "Adrish Banerjee" is displayed. Underneath the name, the affiliation "Department of Electrical Engineering, Indian Institute of Technology Kanpur, Kanpur, Uttar Pradesh, India" is listed. At the bottom of the slide, the date "Apr. 4, 2016" is shown.

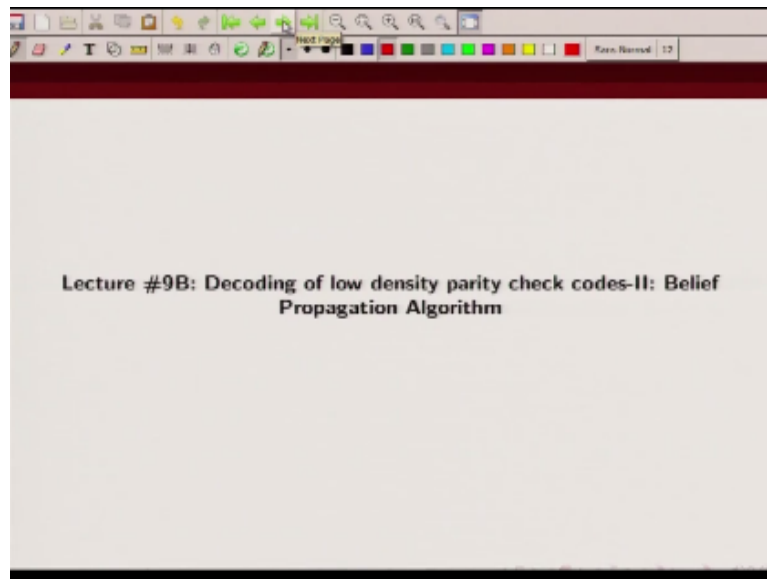
Error control coding: An introduction to linear block codes

Adrish Banerjee

Department of Electrical Engineering  
Indian Institute of Technology Kanpur  
Kanpur, Uttar Pradesh  
India

Apr. 4, 2016

(Refer Slide Time: 00:25)



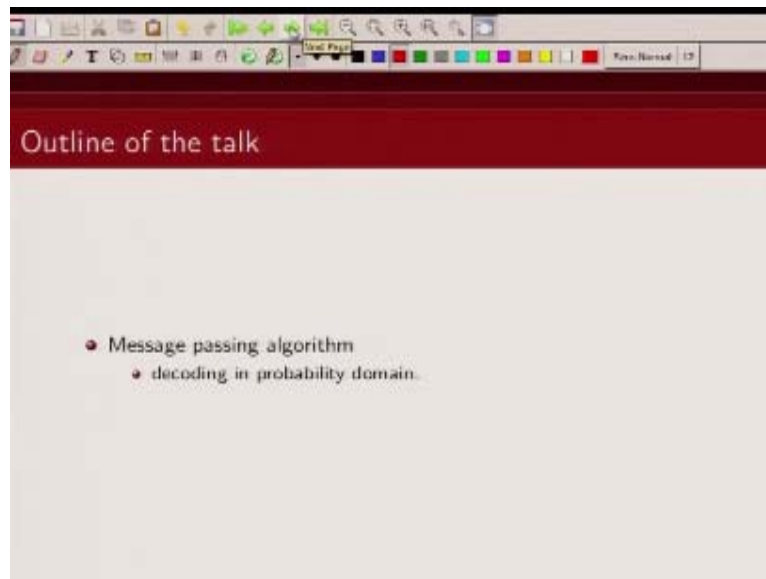
We will continue our discussion on decoding of LDPC codes. Today we are going to talk about probabilistic decoding algorithm.

(Refer Slide Time: 00:32)



So we are going to talk about belief propagation algorithm.

(Refer Slide Time: 00:38)



So before we do that we are going to prove some results and then use those results for decoding of LDPC codes.

(Refer Slide Time: 00:46)

Probabilistic decoding

Theorem:

- Consider a sequence of  $m$  independent random variables  $\mathbf{A} = [A_1, A_2, \dots, A_m]$ , where  $P(A_k = 1) = p_k$ . Then:

$$P(\mathbf{A} \text{ has even parity}) = \frac{1}{2} + \frac{1}{2} \prod_{k=1}^m (1 - 2p_k)$$

and

$$P(\mathbf{A} \text{ has odd parity}) = \frac{1}{2} - \frac{1}{2} \prod_{k=1}^m (1 - 2p_k).$$

So we are going to use this lemma theorem to prove the decoding algorithm results. So what is this result which says if you have a sequence of  $m$  independent random variables which denote by  $\mathbf{A}$ , So  $A_1, A_2, A_3, \dots, A_m$  are  $m$  independent random variables. And probability of  $A_k$  being 1 is given by  $p_k$ , then probability that  $\mathbf{A}$  has even parity is given by this expression. And probability that  $\mathbf{A}$  has odd parity is given by this expression. Now we will use, we will derive our expression for decoding algorithm based on these, this result. So let us first try to prove this result.

(Refer Slide Time: 01:46)

**Probabilistic decoding**

- Consider the function  $\prod_{i=1}^m (1 + P_i t)$
- The coefficient of  $t^i$  is the probability of  $i$ 's.
- The function  $\prod_{i=1}^m (1 - P_i t)$  is identical except for the fact that all odd powers of  $t$  are negative.
- Adding these two functions, all even powers of  $t$  double up and odd powers cancel each other.
- Letting  $t = 1$ , and dividing by 2 we get the probability of getting even ones.

$$P(\mathbf{A} \text{ has even parity}) = \frac{1}{2} + \frac{1}{2} \prod_{i=1}^m (1 - 2p_i)$$

- Similarly we can prove

$$P(\mathbf{A} \text{ has odd parity}) = \frac{1}{2} - \frac{1}{2} \prod_{i=1}^m (1 - 2p_i).$$

So let us consider a function of the form this  $1 - P_i + P_i t$ , so if we look at the co-efficient of  $t$  here this will give us the probability of  $t$  i's. Now we can similarly consider this function where this is, this  $+$  has been replaced by  $-$  here. So this function is identical to this, except that the odd powers of  $t$  will be, have  $-$  in expansion. So if we expand this and we expand this, if we add both of them together what we will get is all the odd powers of  $t$  will go away okay.

So what we will be left with this even powers of  $t$  which are doubled up, and odd powers have cancelled out. Now if you put  $t=1$  and divide by 2 we will get essentially probability that  $A$  has even parity, because we are left with only powers of  $t$  which is even.

(Refer Slide Time: 03:07)

**Probabilistic decoding**

- Consider the function  $\prod_{i=1}^m (1 + P_i t)$
- The coefficient of  $t^i$  is the probability of  $i$ 's.
- The function  $\prod_{i=1}^m (1 - P_i t)$  is identical except for the fact that all odd powers of  $t$  are negative.
- Adding these two functions, all even powers of  $t$  double up and odd powers cancel each other.
- Letting  $t = 1$ , and dividing by 2 we get the probability of getting even ones.

$$P(\mathbf{A} \text{ has even parity}) = \frac{1}{2} + \frac{1}{2} \prod_{i=1}^m (1 - 2p_i)$$

- Similarly we can prove

$$P(\mathbf{A} \text{ has odd parity}) = \frac{1}{2} - \frac{1}{2} \prod_{i=1}^m (1 - 2p_i).$$

And similarly we can also prove probability that A has odd parity. Now in this case this will have all terms positive, odd powers also positive, this will have odd powers negative. So if we subtract this from this, we will get basically the probability of odd parity and that is basically given by this.



(Refer Slide Time: 03:28)

**Probabilistic decoding**

*m+1 → even parity*  
*m even m+1 = 0*  
*m odd m+1 = 1*  
*m+1*

Theorem:  
 Consider a sequence of  $m$  independent random variables  $\mathbf{A} = [A_1, A_2, \dots, A_m]$ , where  $P(A_k = 1) = p_k$ . Then

*m+1 → odd parity*  
*m → odd parity*  
*m → even parity and*  
*or*  
*m → even parity*  
*m+1 → 1*

$$P(\mathbf{A} \text{ has even parity}) = \frac{1}{2} + \frac{1}{2} \prod_{k=1}^m (1 - 2p_k)$$

$$P(\mathbf{A} \text{ has odd parity}) = \frac{1}{2} - \frac{1}{2} \prod_{k=1}^m (1 - 2p_k)$$

*m=1*  
 $= \frac{1}{2} + \frac{1}{2}(1 - 2p_1)$   
 $= 1 - p_1$

*m+1*  
 $(1 - p_{m+1})$   
 $p_{m+1}$   
 $\frac{1}{2} + \frac{1}{2} \prod_{k=1}^{m+1} (1 - 2p_k)$

Now the same method can also be proved using mathematical induction. So if you want to prove it using mathematical induction let us just say we take  $m=1$ . If we take  $m=1$  what do we get here, for  $m=1$  we get basically  $\frac{1}{2} + \frac{1}{2}(1-2P)$ . So this is nothing but  $1-P_1$ . And what is  $P_1$ ?  $P_1$  is the probability, it is 1, so what is the probability it is even parity, it is  $1-P_1$ . So this relation holds for  $m=1$ .

Now let us see it holds for also some  $m$ , then we have to show that it also holds for  $m+1$ . Now if it holds for  $m$  to show that it holds for  $m+1$  we have to find what is the probability that sequence of  $m+1$  independent random variables have even parity. Now when will the  $m+1$  independent random variables will have even parity? They will have even parity when  $m$  of them have even parity and the  $m$ th bit that we get,  $m$ th with this 0.

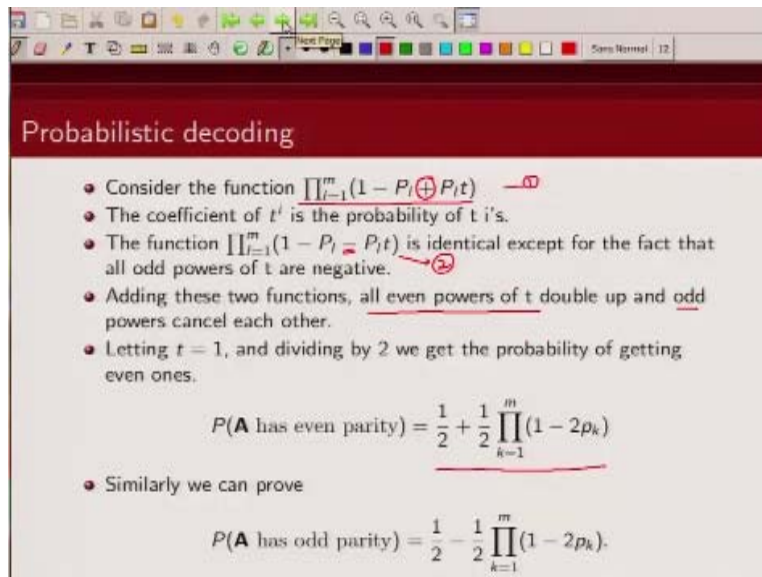
Or some of  $m$  independent random variables they give odd parity and the  $m$ th bit that we receive is actually 1. So then we will get  $m+1$  as even parity, so is it clear? So there are two cases, two ways in which we can get even parity when we are considering  $m+1$  independent random variables. Either  $m$  of them had even parity and  $m+1$  is actually 0, or  $m$  of them add up to odd parity and  $m+1$  is also odd parity, okay.

So what is this probability of  $m$  being even, that is given by this. What is the probability that  $m+1$  is 0 that is given by  $1-P_{m+1}$ . What is the probability that  $m$  of them is odd, that is given by this probability. And what is the probability that  $m+1$  random variable is 1, that probability is given by  $P_{m+1}$ . So the overall probability will be this multiplied by this plus this multiplied by  $P_{m+1}$ .

And if you simplify this we can show that this plus this will come out to be  $1 + \prod_{k=1}^{m+1} (1 - 2P_k)$ . So details of the calculation I am just leaving it, but this is how using mathematical induction we are going to prove that this also holds true for  $m+1$  and hence proof. Similarly we can prove that probability that  $A$  has odd parity is given by this expression. We will first show that for  $m=1$  this is nothing but  $P_1$ .

So that it holds true for  $m=1$ , assuming it holds for  $m$  then we have to show it also holds for  $m+1$ . Now for  $m+1$  to have odd parity 2 ways,  $m$  has odd parity and the  $m+1$  bit that we get is even parity is 0 basically is 0, or  $m$  has even parity and  $m+1$  bit that we get is actually 1. So similar to this case we can also find out these probability and if we add them up we can show that this will be equal to half minus half product from  $k=1$  to  $m+1$  of this, so we can say that  $P(A)$  has even parity is given by this and  $P$  that  $A$  has so some of these  $m$  random variables have odd parity is given by this, so these are the crucial expressions that we will be using in our decoding algorithm.

(Refer Slide Time: 08:38)



### Probabilistic decoding

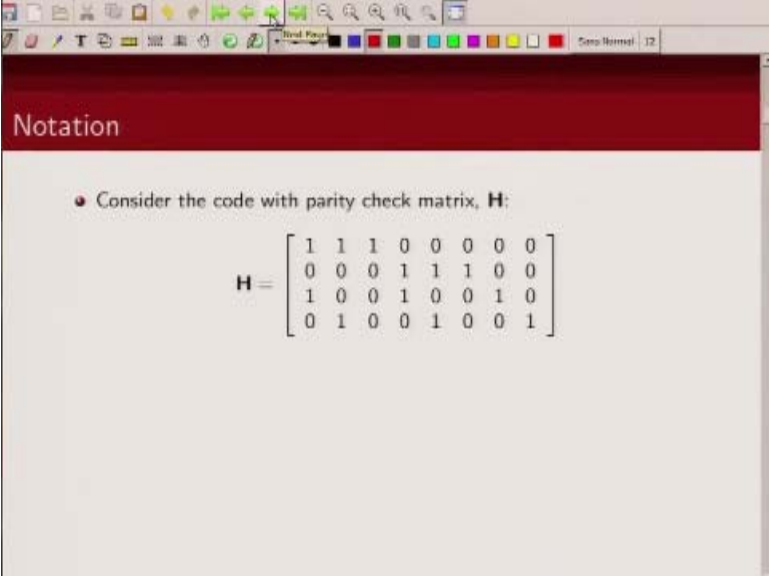
- Consider the function  $\prod_{i=1}^m (1 - P_i \oplus P_i t)$  — (1)
- The coefficient of  $t^i$  is the probability of  $i$ 's.
- The function  $\prod_{i=1}^m (1 - P_i \ominus P_i t)$  is identical except for the fact that all odd powers of  $t$  are negative. — (2)
- Adding these two functions, all even powers of  $t$  double up and odd powers cancel each other.
- Letting  $t = 1$ , and dividing by 2 we get the probability of getting even ones.

$$P(\mathbf{A} \text{ has even parity}) = \frac{1}{2} + \frac{1}{2} \prod_{k=1}^m (1 - 2p_k)$$

- Similarly we can prove

$$P(\mathbf{A} \text{ has odd parity}) = \frac{1}{2} - \frac{1}{2} \prod_{k=1}^m (1 - 2p_k).$$

(Refer Slide Time: 08:40)



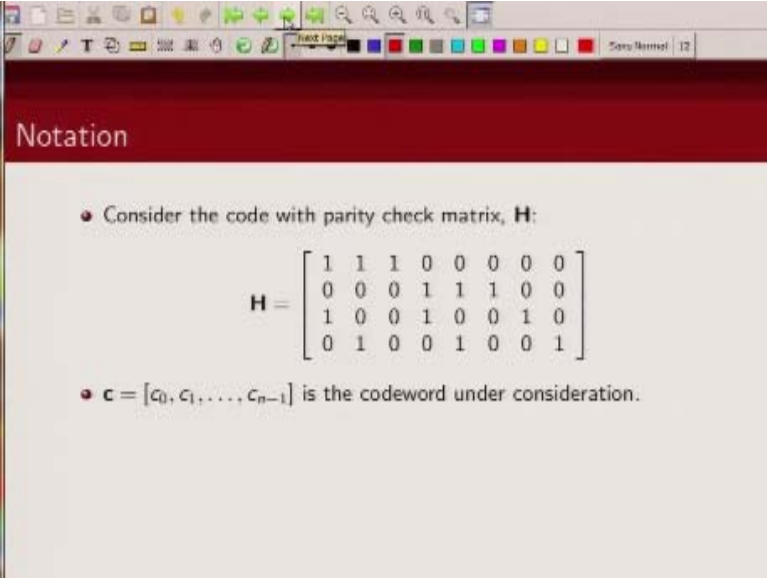
Notation

- Consider the code with parity check matrix, **H**:

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

So let us consider an example, so this our parity check matrix we

(Refer Slide Time: 08:46)



The slide is titled "Notation" in a red header. It contains two bullet points. The first bullet point says "Consider the code with parity check matrix,  $\mathbf{H}$ :" followed by the matrix  $\mathbf{H} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$ . The second bullet point says " $\mathbf{c} = [c_0, c_1, \dots, c_{n-1}]$  is the codeword under consideration."

Notation

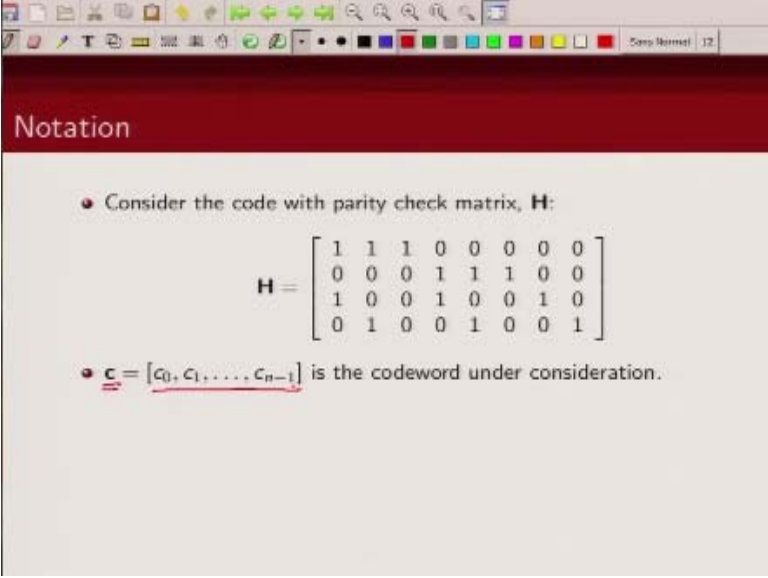
- Consider the code with parity check matrix,  $\mathbf{H}$ :

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

- $\mathbf{c} = [c_0, c_1, \dots, c_{n-1}]$  is the codeword under consideration.

Will first describe the notations that we are going to use and then we will state the decoding algorithm and the corresponding equations and then we illustrate using one example.

(Refer Slide Time: 09:02)



Notation

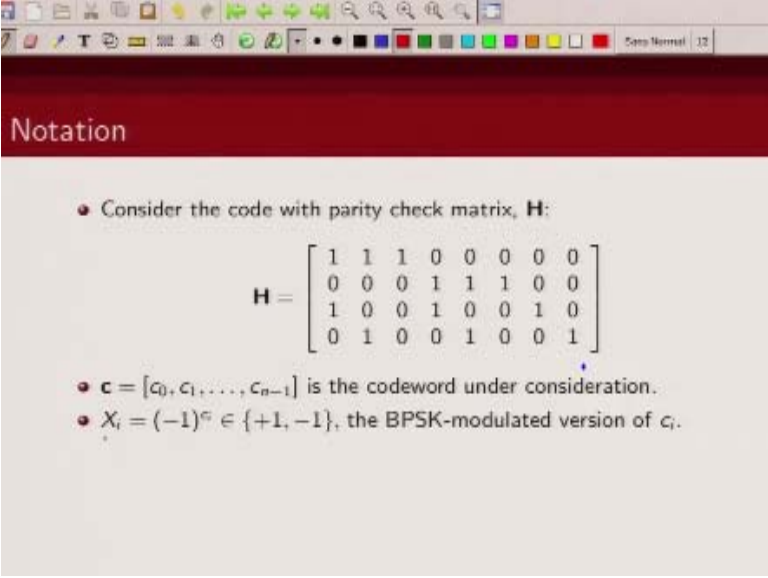
- Consider the code with parity check matrix,  $\mathbf{H}$ :

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

- $\mathbf{c}$  =  $[c_0, c_1, \dots, c_{n-1}]$  is the codeword under consideration.

So we are using  $\mathbf{c}$  to denote our code word, so this is an  $n$  bit code word,  $c_0, c_1, c_2, \dots, c_n$ , that is our code word .

(Refer Slide Time: 09:12)



### Notation

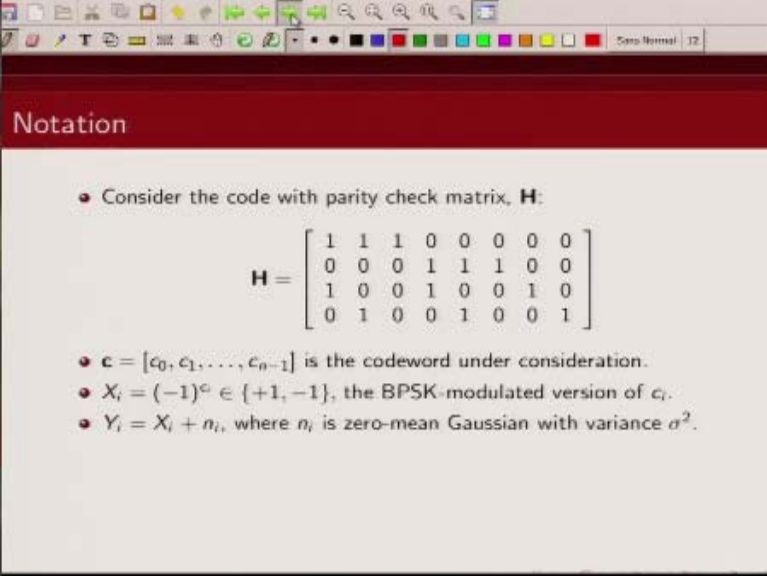
- Consider the code with parity check matrix,  $\mathbf{H}$ :

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

- $\mathbf{c} = [c_0, c_1, \dots, c_{n-1}]$  is the codeword under consideration.
- $X_i = (-1)^{c_i} \in \{+1, -1\}$ , the BPSK-modulated version of  $c_i$ .

Now

(Refer Slide Time: 09:16)



The slide is titled "Notation" and contains the following content:

- Consider the code with parity check matrix,  $\mathbf{H}$ :

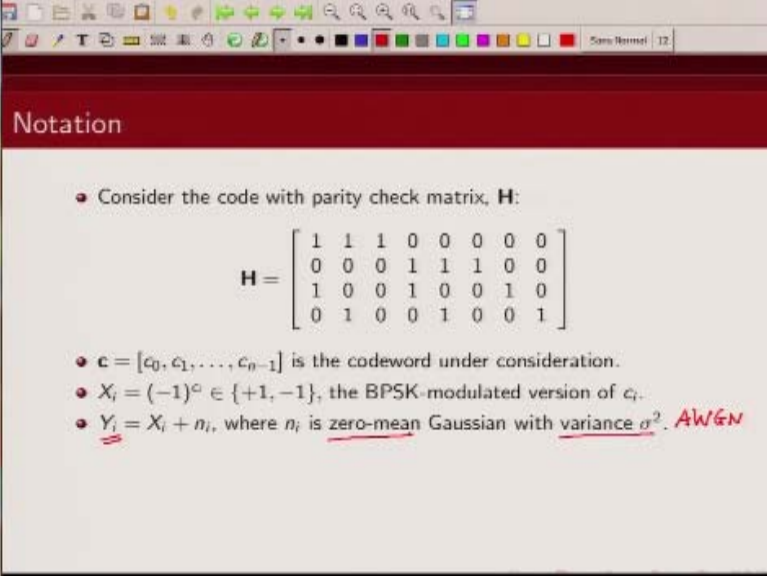
$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

- $\mathbf{c} = [c_0, c_1, \dots, c_{n-1}]$  is the codeword under consideration.
- $X_i = (-1)^{c_i} \in \{+1, -1\}$ , the BPSK-modulated version of  $c_i$ .
- $Y_i = X_i + n_i$ , where  $n_i$  is zero-mean Gaussian with variance  $\sigma^2$ .

This code word is modulated using BPSK modulation so we are mapping 0 to 1 we are mapping to -1, so that is your BPSK modulated version of the code bits denoted by  $X_i$



(Refer Slide Time: 09:32)



The slide is titled "Notation" in a red header. It contains a list of definitions and a matrix  $H$ .

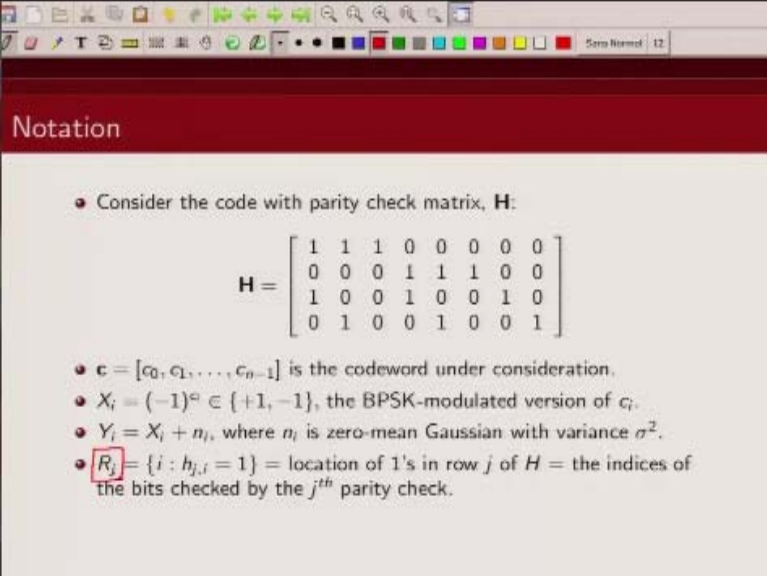
- Consider the code with parity check matrix,  $H$ :

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

- $\mathbf{c} = [c_0, c_1, \dots, c_{n-1}]$  is the codeword under consideration.
- $X_i = (-1)^{c_i} \in \{+1, -1\}$ , the BPSK-modulated version of  $c_i$ .
- $Y_i = X_i + n_i$ , where  $n_i$  is zero-mean Gaussian with variance  $\sigma^2$ . *AWGN*

$Y_i$  is your received modulated code word so this is we are considering an additive white Gaussian noise channel with noise variance given by  $\sigma^2$  and this is zero-mean Gaussian noise

(Refer Slide Time: 09:56)



The slide is titled "Notation" in a red header. It contains a list of definitions and a matrix  $H$ .

- Consider the code with parity check matrix,  $H$ :

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

- $\mathbf{c} = [c_0, c_1, \dots, c_{n-1}]$  is the codeword under consideration.
- $X_i = (-1)^{c_i} \in \{+1, -1\}$ , the BPSK-modulated version of  $c_i$ .
- $Y_i = X_i + n_i$ , where  $n_i$  is zero-mean Gaussian with variance  $\sigma^2$ .
- $R_j = \{i : h_{j,i} = 1\}$  = location of 1's in row  $j$  of  $H$  = the indices of the bits checked by the  $j^{\text{th}}$  parity check.

We use this notation  $R_j$  to denote the location of 1's in row  $j$ . Now what does location of 1's in the rows of the parity check matrix denote? It denotes the bits that are participating in the parity check equation, so let us take

(Refer Slide Time: 10:22)

Notation

- Consider the code with parity check matrix,  $H$ :

$$H = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

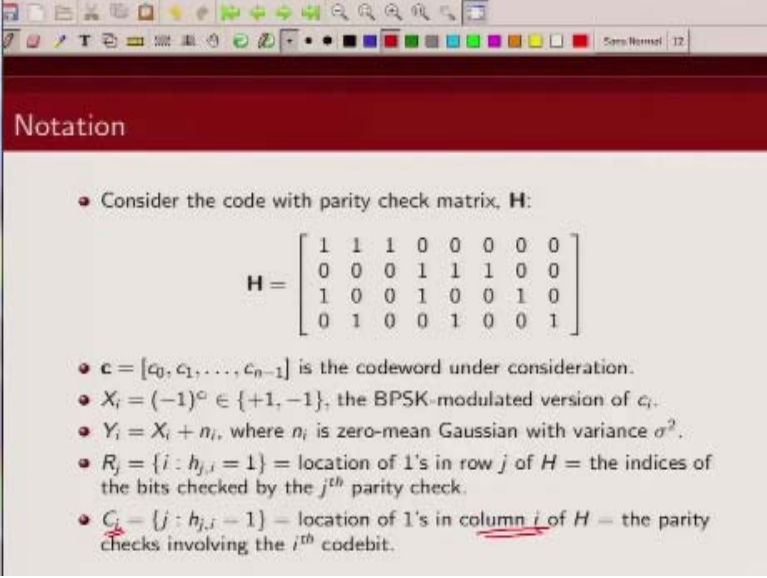
Handwritten notes:

- $R_0 = \{0, 1, 2\}$
- $R_1 = \{3, 4, 5\}$
- $R_2 = \{0, 3, 6\}$
- $R_3 = \{1, 4, 7\}$

- $\mathbf{c} = [c_0, c_1, \dots, c_{n-1}]$  is the codeword under consideration.
- $X_i = (-1)^{c_i} \in \{+1, -1\}$ , the BPSK-modulated version of  $c_i$ .
- $Y_i = X_i + n_i$ , where  $n_i$  is zero-mean Gaussian with variance  $\sigma^2$ .
- $R_j = \{i : h_{j,i} = 1\}$  = location of 1's in row  $j$  of  $H$  = the indices of the bits checked by the  $j^{\text{th}}$  parity check.

This example, what is  $R_0$ , now  $R_0$ , so what are, so  $R_0$  correspond to the 0<sup>th</sup> row which is this row. Now look at the bits that are 1 here 1,2,3 these are bits which are 1, so  $R_0$  corresponds to let us just label them 0,1,2,3,4,5,6, so  $R_0$  corresponds to 0, 1, and 2 and what does it mean, it means for the first parity check equation the bit number 0, 1, and 2 are participating. Similarly  $R_1$  will be this, this, this, so  $R_1$  will be 3, 4, and 5,  $R_2$  will be 0, 3, and 6, and  $R_3$  will be 1, 4, and 7, so that is your  $R_j$

(Refer Slide Time: 11:42)



**Notation**

- Consider the code with parity check matrix,  $\mathbf{H}$ :

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

- $\mathbf{c} = [c_0, c_1, \dots, c_{n-1}]$  is the codeword under consideration.
- $X_i = (-1)^{c_i} \in \{+1, -1\}$ , the BPSK-modulated version of  $c_i$ .
- $Y_i = X_i + n_i$ , where  $n_i$  is zero-mean Gaussian with variance  $\sigma^2$ .
- $R_j = \{i : h_{j,i} = 1\}$  = location of 1's in row  $j$  of  $\mathbf{H}$  = the indices of the bits checked by the  $j^{\text{th}}$  parity check.
- $C_i = \{j : h_{j,i} = 1\}$  = location of 1's in column  $i$  of  $\mathbf{H}$  = the parity checks involving the  $i^{\text{th}}$  codebit.

Now we use this notation  $C_i$  to denote the location of 1's in column  $i$ , now what does location in column  $i$  denotes? It denotes that, that  $i^{\text{th}}$  bit is participating in how many parity check equation, which parity check equation, so let us

(Refer Slide Time: 12:07)

**Notation**

- Consider the code with parity check matrix,  $H$ :

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Handwritten notes above the matrix:

- $C_0 = \{0, 2\}$
- $C_1 = \{0, 3\}$
- $C_4 = \{1, 3\}$

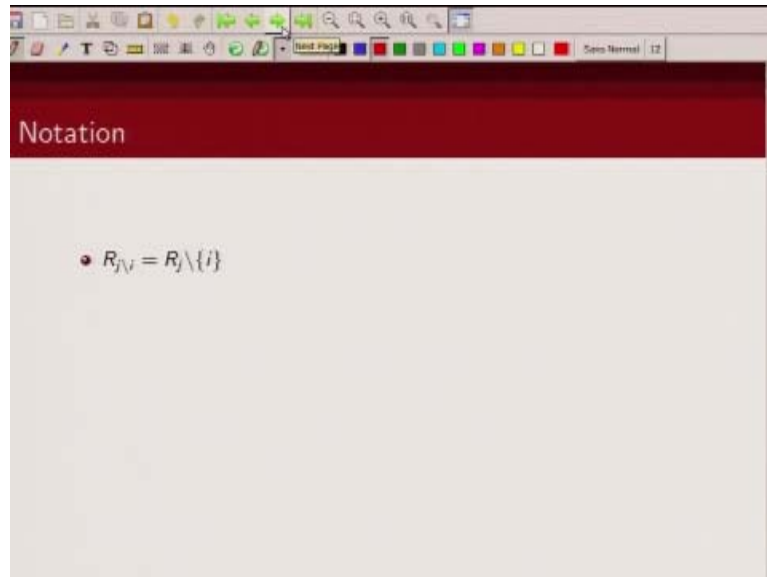
Handwritten arrows pointing to the matrix elements:

- Row 0: 1, 1, 1, 0, 0, 0, 0, 0
- Row 1: 0, 0, 0, 1, 1, 1, 0, 0
- Row 2: 1, 0, 0, 1, 0, 0, 1, 0
- Row 3: 0, 1, 0, 0, 1, 0, 0, 1

- $\mathbf{c} = [c_0, c_1, \dots, c_{n-1}]$  is the codeword under consideration.
- $X_i = (-1)^{c_i} \in \{+1, -1\}$ , the BPSK-modulated version of  $c_i$ .
- $Y_i = X_i + n_i$ , where  $n_i$  is zero-mean Gaussian with variance  $\sigma^2$ .
- $R_j = \{i : h_{j,i} = 1\}$  = location of 1's in row  $j$  of  $H$  = the indices of the bits checked by the  $j^{\text{th}}$  parity check.
- $C_i = \{j : h_{j,i} = 1\}$  = location of 1's in column  $i$  of  $H$  = the parity checks involving the  $i^{\text{th}}$  codebit.

Look at what is  $C_0$ , so this was again 0, 1, 2, 3, 4, 5, 6, 7. So  $C_0$  is what,  $C_0$  is again this is just call it 0, 1, 2, and 3, so this 0<sup>th</sup> parity check equation 1, 2, and 3, so bit number so if we look at column 0 so you can see this bit is participating in 0<sup>th</sup> parity check equation and second parity check equation  $C_0$  will be 0 and 2. Similarity  $C_1$  is going to be 0 because it is participating in 0<sup>th</sup> parity check equation and 3 okay, you can take any let us take  $C_4$  what is  $C_4$ , 4<sup>th</sup> bit is participating in parity check equation 1 and participating in parity check equation 3 so like that you can find out what is  $C_i$ , fine?

(Refer Slide Time: 13:22)



Now we define this set  $R_j$ - a particular element  $i$  so

(Refer Slide Time: 13:31)

**Notation**

- Consider the code with parity check matrix,  $H$ :

$$H = \begin{bmatrix} \underline{1} & \underline{1} & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \underline{1} & 1 & 0 & 0 \\ \underline{1} & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & \underline{1} & 0 & 0 & \underline{1} & 0 & 0 & \underline{1} \end{bmatrix}$$

Handwritten notes above the matrix:

- $C_0 = \{0, 2\}$
- $C_1 = \{0, 3\}$
- $C_2 = \{2, 3\}$

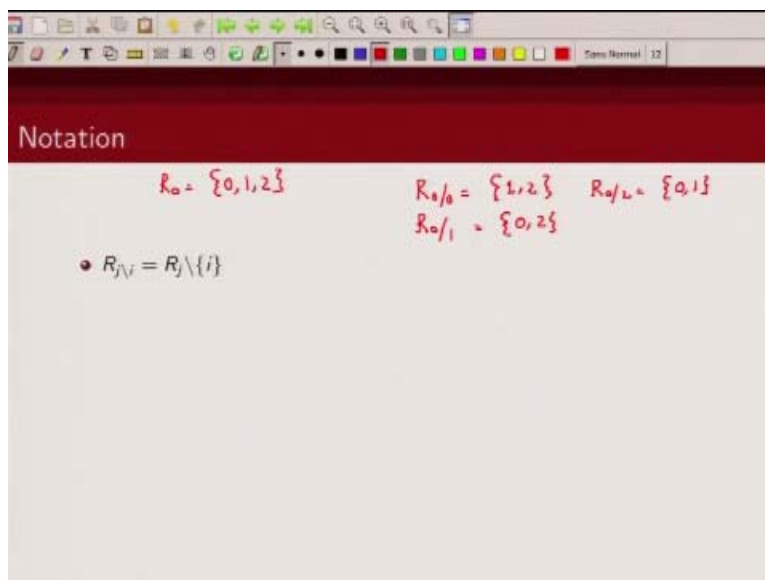
Handwritten annotations on the matrix:

- Red arrows pointing from the first row to columns 0, 1, and 2.
- Red arrows pointing from the second row to columns 3, 4, and 5.
- Red arrows pointing from the third row to columns 0, 3, and 6.
- Red arrows pointing from the fourth row to columns 1, 4, and 7.

- $\mathbf{c} = [c_0, c_1, \dots, c_{n-1}]$  is the codeword under consideration.
- $X_i = (-1)^{c_i} \in \{+1, -1\}$ , the BPSK-modulated version of  $c_i$ .
- $Y_i = X_i + n_i$ , where  $n_i$  is zero-mean Gaussian with variance  $\sigma^2$ .
- $R_j = \{i : h_{j,i} = 1\}$  = location of 1's in row  $j$  of  $H$  = the indices of the bits checked by the  $j^{\text{th}}$  parity check.
- $C_i = \{j : h_{j,i} = 1\}$  = location of 1's in column  $i$  of  $H$  = the parity checks involving the  $i^{\text{th}}$  codebit.

Let us go back, so what was our  $R$  let us say  $R_0$  what was  $R_0$ ?  $R_0$  was location of 1's in 0<sup>th</sup> row, so that location was 0, 1, and 2, right, so if

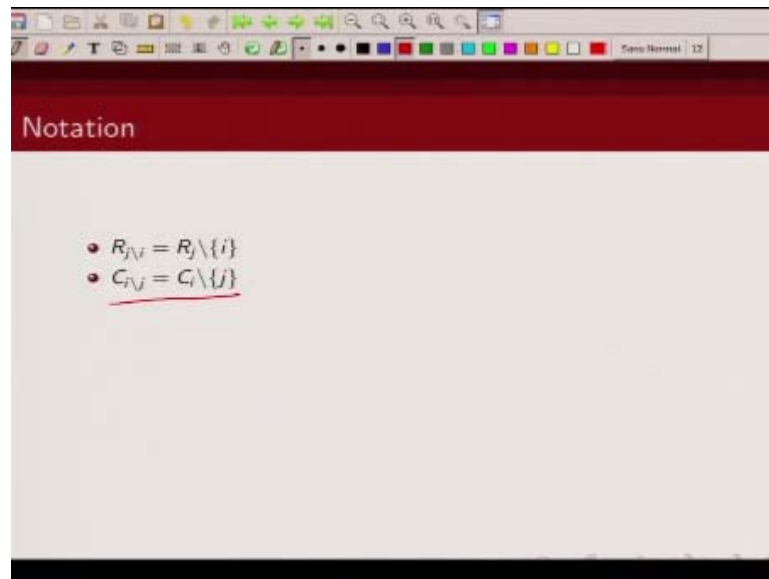
(Refer Slide Time: 13:54)



I define let us say  $R_0 / 0$  that would be then because  $R_0$ ,  $R_0$  is what,  $R_0$  is 0, 1, 2, so  $R_0 - 0$  will have 1 and 2. Similarly  $R_0 - 1$  this will be set containing 0 and 2, and this will be set containing 0 and 1.



(Refer Slide Time: 14:26)



Similarly we define the set  $C_i$  – this element  $j$  so for example

(Refer Slide Time: 14:35)

**Notation**

- Consider the code with parity check matrix,  $H$ :

$$H = \begin{bmatrix} \underline{1} & \underline{1} & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \underline{1} & 1 & 0 & 0 \\ \underline{1} & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & \underline{1} & 0 & 0 & \underline{1} & 0 & 0 & \underline{1} \end{bmatrix}$$

Handwritten notes above the matrix:

- $C_0 = \{0, 2\}$
- $C_1 = \{0, 3\}$
- $C_2 = \{1, 3\}$

Handwritten annotations on the matrix:

- Red arrows pointing from the first row to columns 0, 1, and 2.
- Red arrows pointing from the second row to columns 3, 4, and 5.
- Red arrows pointing from the third row to columns 0, 3, and 6.
- Red arrows pointing from the fourth row to columns 1, 4, and 7.

- $\mathbf{c} = [c_0, c_1, \dots, c_{n-1}]$  is the codeword under consideration.
- $X_i = (-1)^{c_i} \in \{+1, -1\}$ , the BPSK-modulated version of  $c_i$ .
- $Y_i = X_i + n_i$ , where  $n_i$  is zero-mean Gaussian with variance  $\sigma^2$ .
- $R_j = \{i : h_{j,i} = 1\}$  = location of 1's in row  $j$  of  $H$  = the indices of the bits checked by the  $j^{\text{th}}$  parity check.
- $C_i = \{j : h_{j,i} = 1\}$  = location of 1's in column  $i$  of  $H$  = the parity checks involving the  $i^{\text{th}}$  codebit.

In this case let us say  $C_1$  is 0 three

(Refer Slide Time: 14:39)

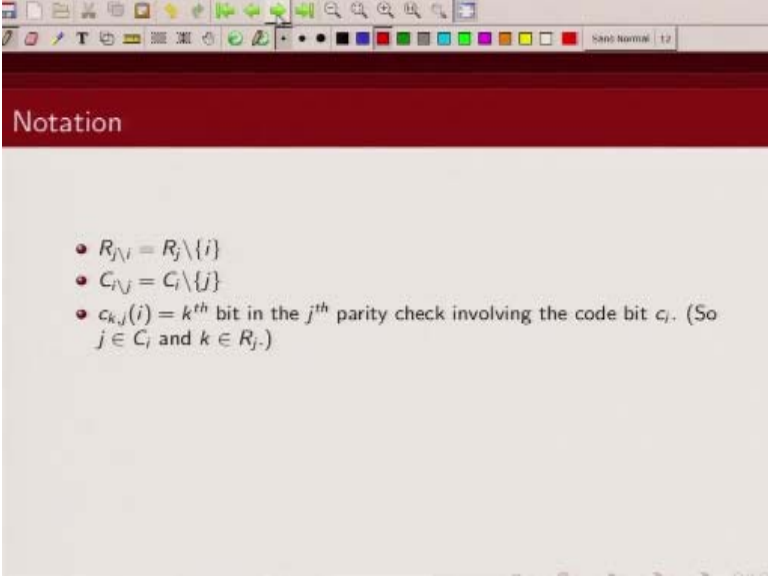
The slide is titled "Notation" and contains the following handwritten mathematical expressions:

- $R_0 = \{0, 1, 2\}$
- $R_0 \setminus 0 = \{1, 2\}$
- $R_0 \setminus 1 = \{0, 2\}$
- $R_0 \setminus 2 = \{0, 1\}$
- $C_1 = \{0, 3\}$
- $C_1 \setminus 0 = \{3\}$
- $C_1 \setminus 3 = \{0\}$

A bullet point on the left side of the slide states:  $R_{j \setminus i} = R_j \setminus \{i\}$

So  $C_1$  has element 0 and 3 so if we define this, this notation is like this, this is this 0 this will be 3 or  $C_1$  will be 0 okay

(Refer Slide Time: 15:06)

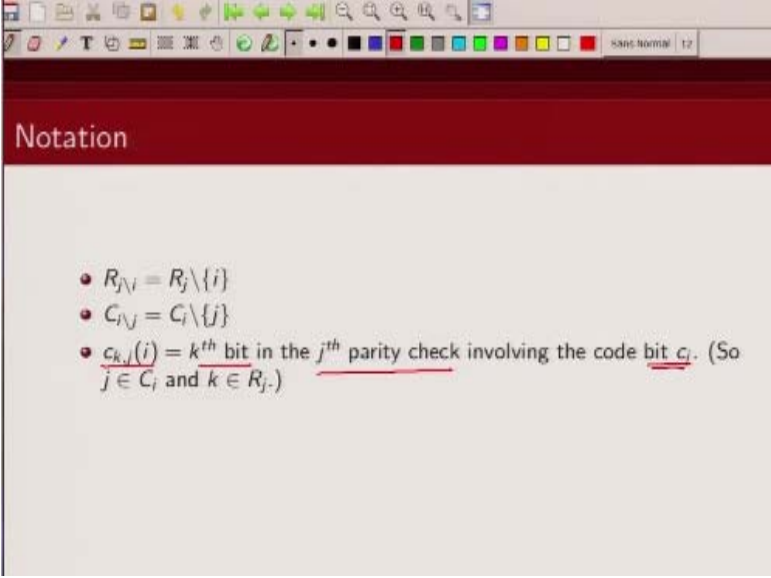


The image shows a presentation slide with a dark red header bar containing the word "Notation" in white. Below the header, on a light beige background, is a bulleted list of three mathematical definitions. The first two definitions use set notation with a backslash to denote the removal of an element from a set. The third definition uses a subscripted notation to describe a specific bit in a parity check, with a note in parentheses clarifying the variables involved.

- $R_{j \setminus i} = R_j \setminus \{i\}$
- $C_{i \setminus j} = C_i \setminus \{j\}$
- $c_{k,j}(i) = k^{\text{th}}$  bit in the  $j^{\text{th}}$  parity check involving the code bit  $c_i$ . (So  $j \in C_i$  and  $k \in R_j$ .)

Now we defined

(Refer Slide Time: 15:10)

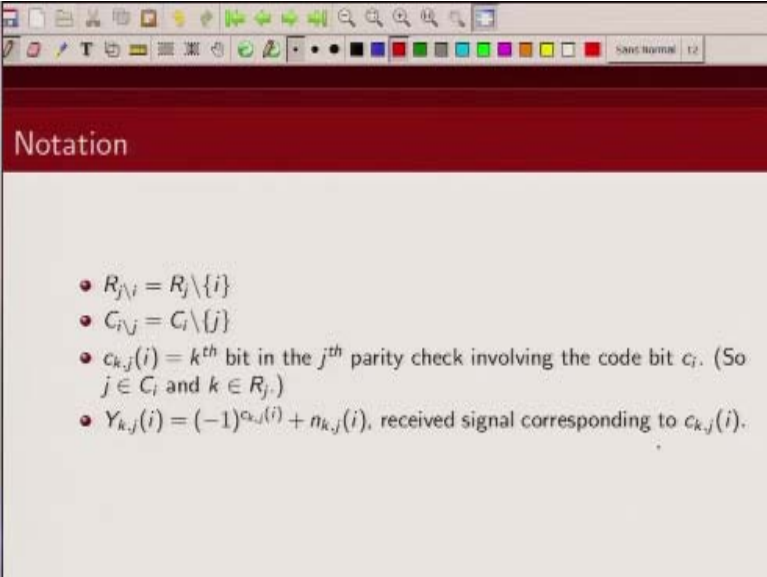


The image shows a presentation slide with a dark red header containing the word "Notation". Below the header, on a light beige background, there is a bulleted list of three mathematical notations. The first two are  $R_{j \setminus i} = R_j \setminus \{i\}$  and  $C_{i \setminus j} = C_i \setminus \{j\}$ . The third is  $c_{k,j}(i) = k^{\text{th}} \text{ bit in the } j^{\text{th}} \text{ parity check involving the code bit } c_i$ . In this third item,  $c_{k,j}(i)$  is underlined, and the phrase "parity check" is also underlined. A note in parentheses follows: "(So  $j \in C_i$  and  $k \in R_{j-}$ )". The slide is displayed within a window that has a standard toolbar at the top and a font selection bar showing "Sans Normal" and size "12".

- $R_{j \setminus i} = R_j \setminus \{i\}$
- $C_{i \setminus j} = C_i \setminus \{j\}$
- $c_{k,j}(i)$  =  $k^{\text{th}}$  bit in the  $j^{\text{th}}$  parity check involving the code bit  $c_i$ . (So  $j \in C_i$  and  $k \in R_{j-}$ )

by  $C_{k,j}$  the  $k^{\text{th}}$  bit in the  $j^{\text{th}}$  parity check equation involving code bit  $c_i$ , that is denoted by  $c_{k,j}(i)$

(Refer Slide Time: 15:24)

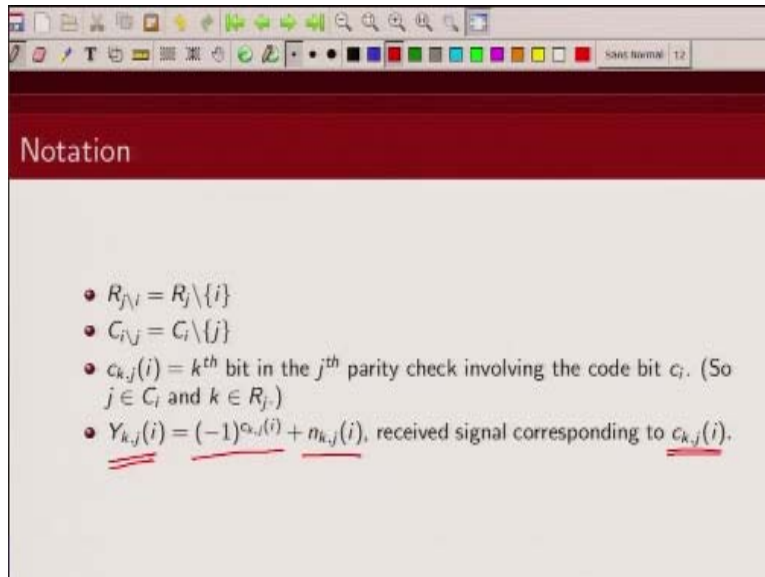


The image shows a presentation slide with a red header bar containing the word "Notation". Below the header, on a light gray background, is a bulleted list of four mathematical definitions. The slide is displayed within a window that has a standard toolbar at the top and a "Save Image" button in the bottom right corner.

- $R_{j \setminus i} = R_j \setminus \{i\}$
- $C_{i \setminus j} = C_i \setminus \{j\}$
- $c_{k,j}(i) = k^{\text{th}}$  bit in the  $j^{\text{th}}$  parity check involving the code bit  $c_i$ . (So  $j \in C_i$  and  $k \in R_j$ .)
- $Y_{k,j}(i) = (-1)^{c_{k,j}(i)} + n_{k,j}(i)$ , received signal corresponding to  $c_{k,j}(i)$ .

So the received sequence corresponding

(Refer Slide Time: 15:24)

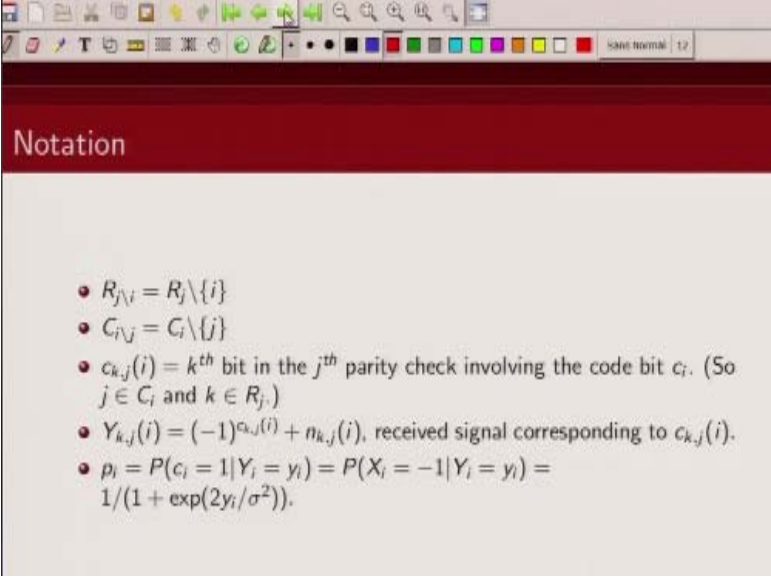


### Notation

- $R_{j \setminus i} = R_j \setminus \{i\}$
- $C_{i \setminus j} = C_i \setminus \{j\}$
- $c_{k,j}(i) = k^{\text{th}}$  bit in the  $j^{\text{th}}$  parity check involving the code bit  $c_i$ . (So  $j \in C_i$  and  $k \in R_{j-}$ )
- $Y_{k,j}(i) = (-1)^{c_{i,j}(i)} + n_{k,j}(i)$ , received signal corresponding to  $c_{k,j}(i)$ .

To this transmitted sequence would be then is the modulated version and this is the noise added.  
So this is the received signal corresponding to this code bit.

(Refer Slide Time: 15:38)



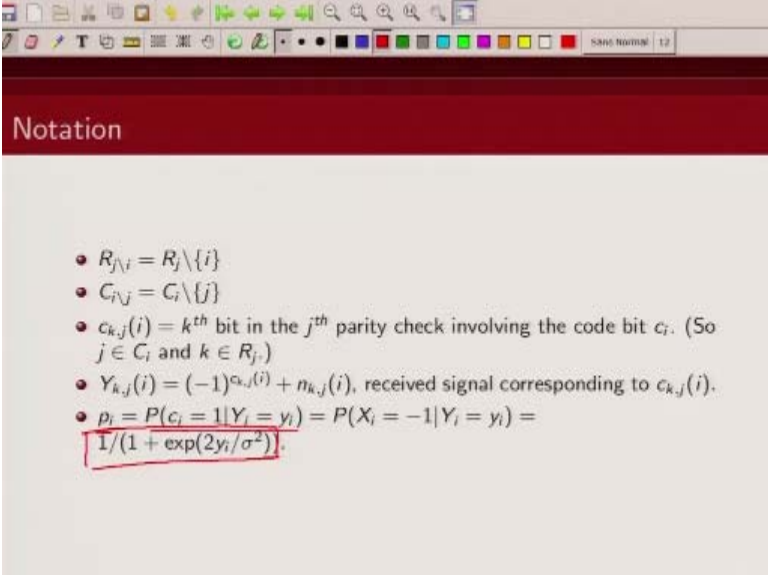
### Notation

- $R_{j \setminus i} = R_j \setminus \{i\}$
- $C_{i \setminus j} = C_i \setminus \{j\}$
- $c_{k,j}(i) = k^{\text{th}}$  bit in the  $j^{\text{th}}$  parity check involving the code bit  $c_i$ . (So  $j \in C_i$  and  $k \in R_j$ .)
- $Y_{k,j}(i) = (-1)^{c_{k,j}(i)} + n_{k,j}(i)$ , received signal corresponding to  $c_{k,j}(i)$ .
- $p_i = P(c_i = 1 | Y_i = y_i) = P(X_i = -1 | Y_i = y_i) = 1/(1 + \exp(2y_i/\sigma^2))$ .

Now for an AWJN channel we can compute this probability



(Refer Slide Time: 15:38)

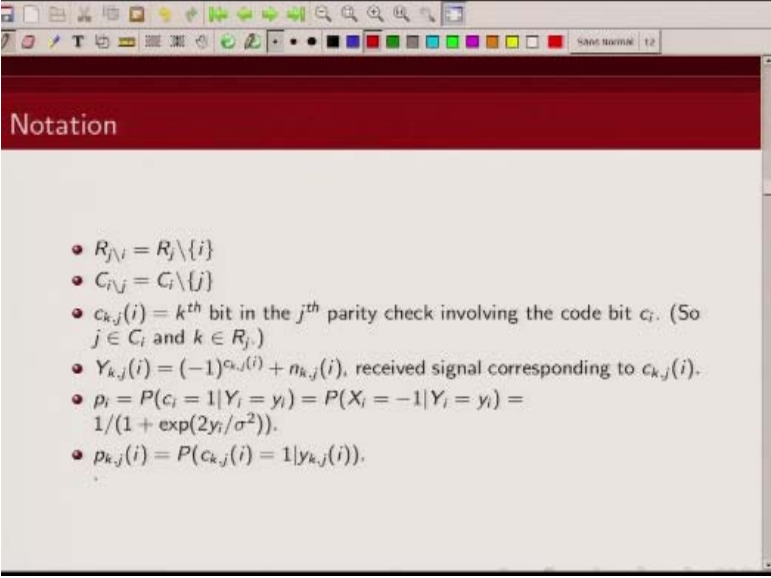


**Notation**

- $R_{j \setminus i} = R_j \setminus \{i\}$
- $C_{i \setminus j} = C_i \setminus \{j\}$
- $c_{k,j}(i)$  =  $k^{\text{th}}$  bit in the  $j^{\text{th}}$  parity check involving the code bit  $c_i$ . (So  $j \in C_i$  and  $k \in R_j$ .)
- $Y_{k,j}(i) = (-1)^{c_{k,j}(i)} + n_{k,j}(i)$ , received signal corresponding to  $c_{k,j}(i)$ .
- $\rho_i = P(c_i = 1 | Y_i = y_i) = P(X_i = -1 | Y_i = y_i) = \frac{1}{1 + \exp(2y_i/\sigma^2)}$ .

That  $P(C_i = 1 \mid Y_i)$ , this can be given by this expression

(Refer Slide Time: 16:00)

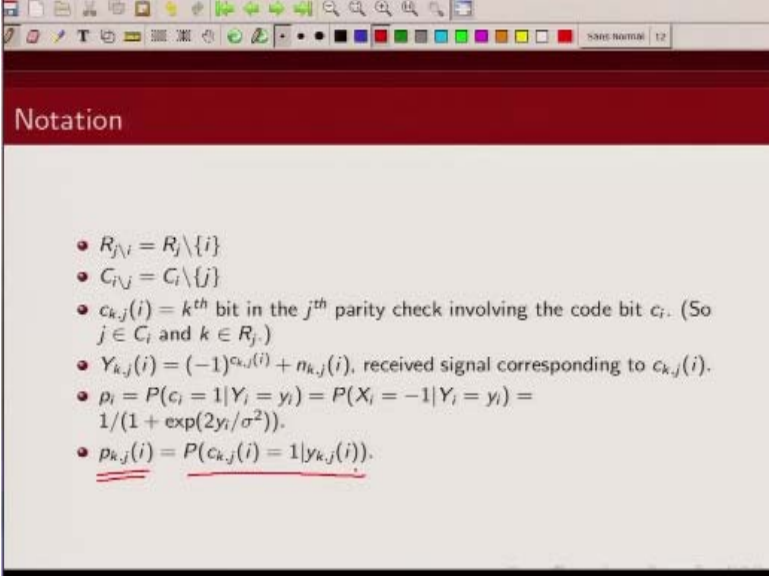


The image shows a presentation slide with a red header bar containing the word "Notation". Below the header, there is a list of mathematical notations and definitions, each preceded by a red circular bullet point. The slide is displayed within a window that has a standard toolbar at the top and a status bar at the bottom showing "Sans Normal" and "12".

- $R_{j \setminus i} = R_j \setminus \{i\}$
- $C_{i \setminus j} = C_i \setminus \{j\}$
- $c_{k,j}(i) = k^{th}$  bit in the  $j^{th}$  parity check involving the code bit  $c_i$ . (So  $j \in C_i$  and  $k \in R_j$ .)
- $Y_{k,j}(i) = (-1)^{c_{k,j}(i)} + n_{k,j}(i)$ , received signal corresponding to  $c_{k,j}(i)$ .
- $\rho_i = P(c_i = 1 | Y_i = y_i) = P(X_i = -1 | Y_i = y_i) = 1 / (1 + \exp(2y_i / \sigma^2))$ .
- $p_{k,j}(i) = P(c_{k,j}(i) = 1 | y_{k,j}(i))$ .

And we denote by

(Refer Slide Time: 16:01)

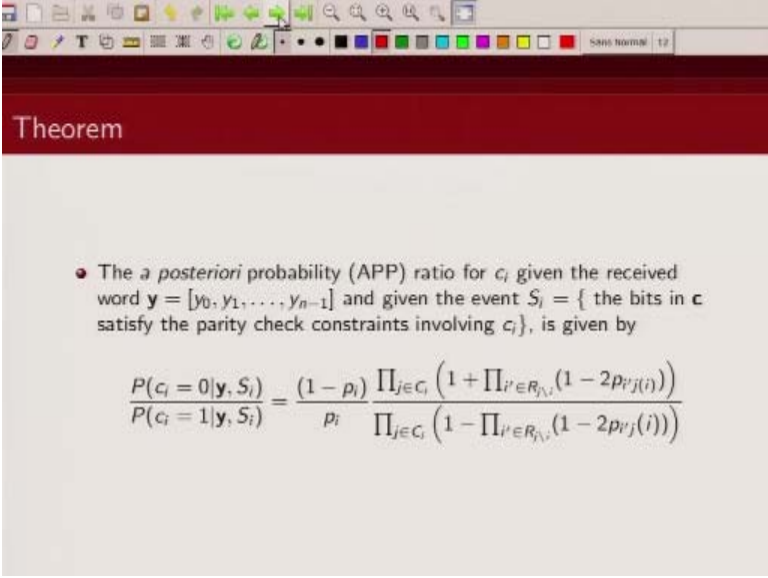


The image shows a presentation slide with a red header bar containing the word "Notation". Below the header, there is a list of six mathematical definitions, each preceded by a red bullet point. The definitions are as follows:

- $R_{j \setminus i} = R_j \setminus \{i\}$
- $C_{i \setminus j} = C_i \setminus \{j\}$
- $c_{k,j}(i) = k^{\text{th}}$  bit in the  $j^{\text{th}}$  parity check involving the code bit  $c_i$ . (So  $j \in C_i$  and  $k \in R_j$ .)
- $Y_{k,j}(i) = (-1)^{c_{k,j}(i)} + n_{k,j}(i)$ , received signal corresponding to  $c_{k,j}(i)$ .
- $\rho_i = P(c_i = 1 | Y_i = y_i) = P(X_i = -1 | Y_i = y_i) = 1/(1 + \exp(2y_i/\sigma^2))$ .
- $p_{k,j}(i) = P(c_{k,j}(i) = 1 | y_{k,j}(i))$ .

$P_{k,j}$  the probability that  $C_{k,j}$  is 1 given a received sequence  $Y_{k,j}$

(Refer Slide Time: 16:08)



Theorem

- The *a posteriori* probability (APP) ratio for  $c_i$  given the received word  $\mathbf{y} = [y_0, y_1, \dots, y_{n-1}]$  and given the event  $S_i = \{ \text{the bits in } \mathbf{c} \text{ satisfy the parity check constraints involving } c_i \}$ , is given by

$$\frac{P(c_i = 0 | \mathbf{y}, S_i)}{P(c_i = 1 | \mathbf{y}, S_i)} = \frac{(1 - p_i)}{p_i} \frac{\prod_{j \in C_i} \left( 1 + \prod_{i' \in R_{j,i'}} (1 - 2p_{i'j(i)}) \right)}{\prod_{j \in C_i} \left( 1 - \prod_{i' \in R_{j,i'}} (1 - 2p_{i'j(i)}) \right)}$$

So then we can write down the expression for a posteriori

(Refer Slide Time: 16:09)

**Theorem**

• The *a posteriori* probability (APP) ratio for  $c_i$  given the received word  $\mathbf{y} = [y_1, y_2, \dots, y_{n-1}]$  and given the event  $S_i = \{ \text{the bits in } \mathbf{c} \text{ satisfy the parity check constraints involving } c_i \}$ , is given by

$$\frac{P(c_i = 0 | \mathbf{y}, S_i)}{P(c_i = 1 | \mathbf{y}, S_i)} = \frac{(1 - p_i) \prod_{j \in C_i} (1 + \prod_{r \in R_{j,i}} (1 - 2p_{r,j}(i)))}{p_i \prod_{j \in C_i} (1 - \prod_{r \in R_{j,i}} (1 - 2p_{r,j}(i)))}$$

Probability of our code bit  $C_i$  given our received sequence is  $\mathbf{y}$  and given that the parity check constraints containing  $C_i$  has been satisfied. So what is the probability of  $C_i$  being 0 given a received sequence  $\mathbf{Y}$  and given that the parity check constrain containing involving  $C_i$  has been satisfied. This is given by divided by probability of  $C_i$  being 1 given  $\mathbf{Y}$  and  $S_i$ , this expression is given by this expression, and we are going to use the theorem that we have proved in the beginning of the lecture to derive this expression, namely if you have  $m$  independent random variables what is the probability that some of them has even parity and some of them, some of them have odd parity, we are going to make use of that result.

(Refer Slide Time: 17:21)

**Theorem**

- The *a posteriori probability (APP)* ratio for  $c_i$  given the received word  $\mathbf{y} = [y_1, y_2, \dots, y_n]$  and given the event  $S_i = \{ \text{the bits in } \mathbf{c} \text{ satisfy the parity check constraints involving } c_i \}$ , is given by

$$\frac{P(c_i = 0 | \mathbf{y}, S_i)}{P(c_i = 1 | \mathbf{y}, S_i)} = \frac{(1 - p_i) \prod_{j \in C_i} (1 + \prod_{r \in R_{ji}} (1 - 2p_{r_j(i)}))}{p_i \prod_{j \in C_i} (1 - \prod_{r \in R_{ji}} (1 - 2p_{r_j(i)}))}$$

To derive this expression.

(Refer Slide Time: 17:25)

Proof

• From Bayes' rule:

$$\frac{P(c_i = 0 | y, S_i)}{P(c_i = 1 | y, S_i)} = \frac{\overbrace{P(c_i = 0 | y_i)}^{1 - p_i} P(S_i | c_i = 0, y)}{\underbrace{P(c_i = 1 | y_i)}_{p_i} P(S_i | c_i = 1, y)}$$

So let us see, so from base rule we can write this probability as probability of  $C_i$  being 0 given  $Y_i$ , multiplied by probability that the parity check constraints are satisfied given  $C_i$  is 0 and the received sequence is  $Y$ , and similarly a denominator we can write, that is probability of  $C_i$  being 1 given receive sequence  $Y$  and multiplied by the probability that parity check constraints are, involving  $C_i$  is satisfied when  $C_i$  is 1 so this, this probability is nothing but our  $P_i$ .

(Refer Slide Time: 18:09)

**Theorem**

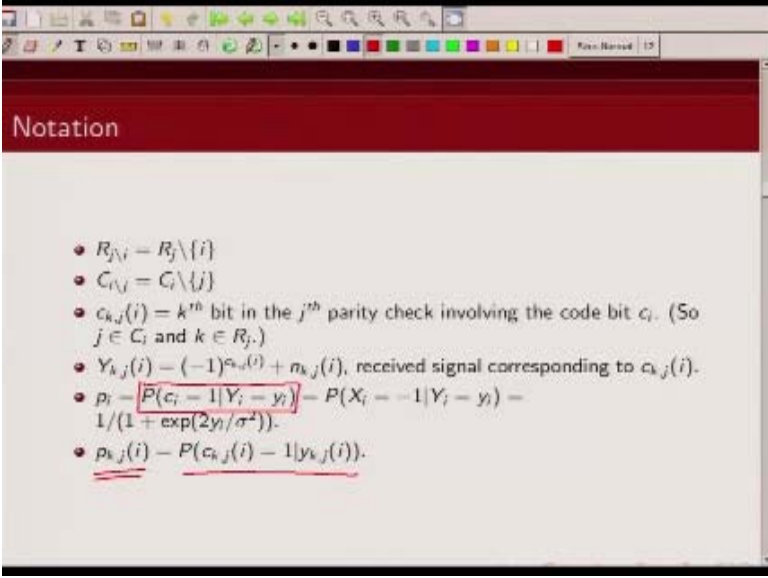
- The *a posteriori probability (APP)* ratio for  $c_i$  given the received word  $\mathbf{y} = [y_1, y_2, \dots, y_{n-1}]$  and given the event  $S_i = \{ \text{the bits in } \mathbf{c} \text{ satisfy the parity check constraints involving } c_i \}$ , is given by

$$\frac{P(c_i = 0 | \mathbf{y}, S_i)}{P(c_i = 1 | \mathbf{y}, S_i)} = \frac{(1 - p_i) \prod_{j \in C_i} (1 + \prod_{r \in R_{ji}} (1 - 2p_{r_j(i)}))}{p_i \prod_{j \in C_i} (1 - \prod_{r \in R_{ji}} (1 - 2p_{r_j(i)}))}$$

To go back.



(Refer Slide Time: 18:10)

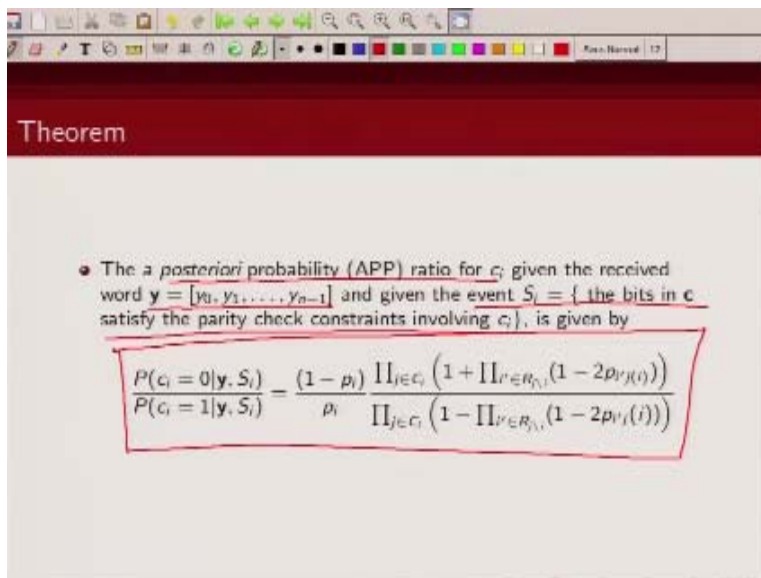


The slide is titled "Notation" and contains the following definitions and formulas:

- $R_{j \setminus i} = R_j \setminus \{i\}$
- $C_{i \setminus j} = C_i \setminus \{j\}$
- $c_{k,j}(i) = k^{\text{th}}$  bit in the  $j^{\text{th}}$  parity check involving the code bit  $c_i$ . (So  $j \in C_i$  and  $k \in R_j$ .)
- $Y_{k,j}(i) = (-1)^{c_{k,j}(i)} + n_{k,j}(i)$ , received signal corresponding to  $c_{k,j}(i)$ .
- $p_i = \frac{P(c_i = 1 | Y_i = y_i)}{1/(1 + \exp(2y_i/\sigma^2))} = P(X_i = -1 | Y_i = y_i) =$
- $\underline{p_{k,j}(i)} = \underline{P(c_{k,j}(i) = 1 | y_{k,j}(i))}$ .

What was  $P_i$ ,  $P_i$  is probability of  $C_i$  given  $V_i$  so that is  $P_i$ .

(Refer Slide Time: 18:16)



The screenshot shows a presentation slide with a red header bar containing the word "Theorem". Below the header, there is a bullet point describing the a posteriori probability (APP) ratio for  $c_i$  given the received word  $\mathbf{y} = [y_1, y_2, \dots, y_{n-1}]$  and the event  $S_i = \{\text{the bits in } \mathbf{c} \text{ satisfy the parity check constraints involving } c_i\}$ . The formula for the APP ratio is enclosed in a red hand-drawn box:

$$\frac{P(c_i = 0 | \mathbf{y}, S_i)}{P(c_i = 1 | \mathbf{y}, S_i)} = \frac{(1 - p_i) \prod_{j \in c_i} (1 + \prod_{l' \in R_{j,i}} (1 - 2p_{l'(i)}))}{p_i \prod_{j \in c_i} (1 - \prod_{l' \in R_{j,i}} (1 - 2p_{l'(i)}))}$$

(Refer Slide Time: 18:17)

Proof

• From Bayes' rule:

$$\frac{P(c_i = 0 | \mathbf{y}, S_i)}{P(c_i = 1 | \mathbf{y}, S_i)} = \frac{\overbrace{P(c_i = 0 | y_i)}^{1 - p_i} P(S_i | c_i = 0, \mathbf{y})}{\underbrace{P(c_i = 1 | y_i)}_{p_i} \underbrace{P(S_i | c_i = 1, \mathbf{y})}_{\text{✓}}}$$

So then this with, this upper term would be  $1 - p_i$  okay now let us pay close attention to these terms then.

(Refer Slide Time: 18:29)

Proof

- From Bayes' rule:

$$\frac{P(c_i = 0 | y, S_i)}{P(c_i = 1 | y, S_i)} = \frac{\overbrace{P(c_i = 0 | y_i) P(S_i | c_i = 0, y)}^{1-p_i}}{\underbrace{P(c_i = 1 | y_i) P(S_i | c_i = 1, y)}_{p_i}}$$

- Let's consider the term  $P(S_i | c_i = 0, y)$ . Given  $c_i = 0$ ,  $S_i$  holds if each of  $w_c$  parity checks involving  $c_i$  has the property that the  $w_c - 1$  bits in the check *other than*  $c_i$  have even parity.

So what is this given that my code bits  $C_i$  is 0, when will parity check constrain involving  $C_i$  will be satisfied? It is when sum of other parity bits involved in the parity check constraints they add up to have even parity right.

(Refer Slide Time: 18:57)

Proof

- From Bayes' rule:

$$\frac{P(c_i = 0 | y, S_i)}{P(c_i = 1 | y, S_i)} = \frac{\overbrace{P(c_i = 0 | y_i)}^{1-p_i} P(S_i | c_i = 0, y)}{\underbrace{P(c_i = 1 | y_i)}_{p_i} P(S_i | c_i = 1, y)}$$

- Let's consider the term  $P(S_i | c_i = 0, y)$ . Given  $c_i = 0$ ,  $S_i$  holds if each of  $w_i$  parity checks involving  $c_i$  has the property that the  $w_i - 1$  bits in the check other than  $c_i$  have even parity.

So this even  $C_i$  will be satisfied if each of these parity check equations where  $C_i$  is involved other than this  $C_i$  bit if all other bits involved in the parity check equation in case of us regularity Pc code that number is  $w_r - 1$ , if those all those bits have even parity because  $C_i$  has  $C_i$  is 0 so the other bits should have even parity then only the parity check equation involving  $C_i$  will be satisfied, so we need to find the condition that sum of other parity check bits involved in the parity check equations where  $C_i$  is participating they should have even parity.

(Refer Slide Time: 20:01)

**Proof**

- From Bayes' rule:

$$\frac{P(c_i = 0 | \mathbf{y}, S_i)}{P(c_i = 1 | \mathbf{y}, S_i)} = \frac{\overbrace{P(c_i = 0 | y_i)}^{1-p_i} P(S_i | c_i = 0, \mathbf{y})}{\underbrace{P(c_i = 1 | y_i)}_{p_i} P(S_i | c_i = 1, \mathbf{y})}$$

- Let's consider the term  $P(S_i | c_i = 0, \mathbf{y})$ . Given  $c_i = 0$ ,  $S_i$  holds if each of  $w_i$  parity checks involving  $c_i$  has the property that the  $w_i - 1$  bits in the check *other than*  $c_i$  have even parity.
- For parity check  $j \in C_i$ , the probability that the  $w_i - 1$  bits other than  $c_i$  have even parity is given by the lemma to be:

$$\frac{1}{2} + \frac{1}{2} \prod_{i' \in R_{j,i}} (1 - 2p_{i',j}(i)).$$

Now from the theorem that we have proved we know what is that probability, probability that other than  $C_i$   $w_i - 1$  bits they have even parity, that probability is given by this expression, if you go back.

(Refer Slide Time: 20:25)

**Probabilistic decoding**

- Consider the function  $\prod_{i=1}^m (1 - P_i t)$  — ①
- The coefficient of  $t^i$  is the probability of  $i$ 's.
- The function  $\prod_{i=1}^m (1 - P_i t)$  is identical except for the fact that all odd powers of  $t$  are negative. — ②
- Adding these two functions, all even powers of  $t$  double up and odd powers cancel each other.
- Letting  $t = 1$ , and dividing by 2 we get the probability of getting even ones.

$$P(\mathbf{A} \text{ has even parity}) = \frac{1}{2} + \frac{1}{2} \prod_{k=1}^m (1 - 2p_k)$$

- Similarly we can prove

$$P(\mathbf{A} \text{ has odd parity}) = \frac{1}{2} - \frac{1}{2} \prod_{k=1}^m (1 - 2p_k).$$

Probability that they have probability that  $m$  random variables have even parity is  $\frac{1}{2} + \frac{1}{2}$  product  $1 - 2p_k$ , so probability that  $wr = 1$  bit.

(Refer Slide Time: 20:45)

**Proof**

- From Bayes' rule:

$$\frac{P(c_i = 0 | \mathbf{y}, S_i)}{P(c_i = 1 | \mathbf{y}, S_i)} = \frac{\overbrace{P(c_i = 0 | \mathbf{y}_i)}^{1-p_i} P(S_i | c_i = 0, \mathbf{y})}{\underbrace{P(c_i = 1 | \mathbf{y}_i)}_{p_i} P(S_i | c_i = 1, \mathbf{y})}.$$

- Let's consider the term  $P(S_i | c_i = 0, \mathbf{y})$ . Given  $c_i = 0$ ,  $S_i$  holds if each of  $w_i$  parity checks involving  $c_i$  has the property that the  $w_i - 1$  bits in the check *other than*  $c_i$  have even parity.
- For parity check  $j \in C_i$ , the probability that the  $w_i - 1$  bits other than  $c_i$  have even parity is given by the lemma to be:

$$\frac{1}{2} + \frac{1}{2} \prod_{i' \in R_{i,j}} (1 - 2p_{i',j}(i)).$$

$R_j$

So  $\frac{1}{2} + \frac{1}{2}$  now pay close attention to this, this  $1 - 2p_k$  now what are the bits that we are considering, now what will  $R_j$  tell us,  $R_j$  will tell us the  $G_r$  parity check equation.



(Refer Slide Time: 21:04)

**Proof**

- From Bayes' rule:

$$\frac{P(c_i = 0 | y, S_i)}{P(c_i = 1 | y, S_i)} = \frac{\overbrace{P(c_i = 0 | y_i)}^{1-p_i} P(S_i | c_i = 0, y)}{\underbrace{P(c_i = 1 | y_i)}_{p_i} P(S_i | c_i = 1, y)}$$

- Let's consider the term  $P(S_i | c_i = 0, y)$ . Given  $c_i = 0$ ,  $S_i$  holds if each of  $w_i$  parity checks involving  $c_i$  has the property that the  $w_i - 1$  bits in the check *other than*  $c_i$  have even parity.
- For parity check  $j \in C_i$ , the probability that the  $w_i - 1$  bits other than  $c_i$  have even parity is given by the lemma to be:

$$\frac{1}{2} + \frac{1}{2} \prod_{i' \in R_{i,j}} (1 - 2p_{i',j}(i)).$$

$R_{j,i}$

And  $R_j - I$  where  $I$  is,  $C_i$  bit is involved and this is  $-i$ , so the other bits other than  $C_i$  which are participating in the parity check constrain product of this, I mean they should add up to have even parity so this is a set where  $C_i$  is participating in a parity check constrain, so other than  $C_i$ , other bits have even parity that is captured by this particular set.

(Refer Slide Time: 21:35)

Probabilistic Decoding

- The independence of the  $y_i$ 's means that the probability that all  $w_c$  parity checks involving  $c_i$  are satisfied (given  $c_i = 0$ ) is just

$$P(S_i | c_i = 0, \mathbf{y}) = \prod_{j \in C_i} \left( \frac{1}{2} + \frac{1}{2} \prod_{j' \in R_{j,i}} (1 - 2p_{ij'}) \right).$$

And this should hold for all parity check equations involving  $C_i$  so this should hold for all  $w_c$  parity checks sets where this particular bit  $C_i$  is participating, so that is why you assuming that why  $i^s$  are independent I can then write the probability as product over all such parity checks equations where this is involved, so I can write down then probability that my parity check set constrain is satisfied given  $C_i$  is 0 is given by this expression.

(Refer Slide Time: 22:24)

**Probabilistic Decoding**

- The independence of the  $y_i$ 's means that the probability that *all*  $w_c$  parity checks involving  $c_i$  are satisfied (given  $c_i = 0$ ) is just

$$P(S_i | c_i = 0, \mathbf{y}) = \prod_{j \in C_i} \left( \frac{1}{2} + \frac{1}{2} \prod_{j' \in R_{ij}} (1 - 2p_{ij'}(j)) \right).$$

- Similar analysis assuming  $c_i = 1$  yields

$$\underline{P(S_i | c_i = 1, \mathbf{y})} = \prod_{j \in C_i} \left( \frac{1}{2} - \frac{1}{2} \prod_{j' \in R_{ij}} (1 - 2p_{ij'}(j)) \right).$$

And I can follow the same logic to find out the probability when  $C_i$  is 1, when  $C_i$  is 1 what I want the other bits should add up to have an odd parity and that is.

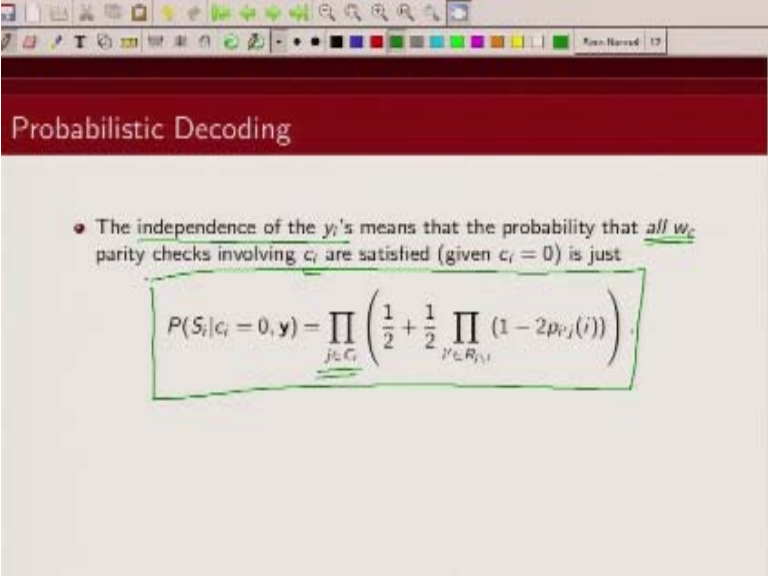
(Refer Slide Time: 22:40)

**Probabilistic Decoding**

- The independence of the  $y_i$ 's means that the probability that *all*  $w_c$  parity checks involving  $c_i$  are satisfied (given  $c_i = 0$ ) is just
 
$$P(S_i | c_i = 0, \mathbf{y}) = \prod_{j \in C_i} \left( \frac{1}{2} + \frac{1}{2} \prod_{j' \in R_{j,i}} (1 - 2p_{ij'}(j)) \right).$$
- Similar analysis assuming  $c_i = 1$  yields
 
$$P(S_i | c_i = 1, \mathbf{y}) = \prod_{j \in C_i} \left( \frac{1}{2} - \frac{1}{2} \prod_{j' \in R_{j,i}} (1 - 2p_{ij'}(j)) \right).$$

Given by this expression and this should hold for all parity check,  $w_c$  parity check equations so this is assuming independence I can multiply by each of these probabilities, so this is the probability of this parity check set, I mean parity check constrain getting satisfied when  $C_i$  is 1 and this is the expression when  $C_i$  is 0. So if I plug these values.

(Refer Slide Time: 23:11)



Probabilistic Decoding

- The independence of the  $y_i$ 's means that the probability that all  $w_c$  parity checks involving  $c_i$  are satisfied (given  $c_i = 0$ ) is just

$$P(S_i | c_i = 0, \mathbf{y}) = \prod_{j \in C_i} \left( \frac{1}{2} + \frac{1}{2} \prod_{j' \in R_{j,i}} (1 - 2p_{ij'}) \right).$$

(Refer Slide Time: 23:12)

**Proof**

- From Bayes' rule:

$$\frac{P(c_i = 0 | \mathbf{y}, S_i)}{P(c_i = 1 | \mathbf{y}, S_i)} = \frac{\overbrace{P(c_i = 0 | \mathbf{y}_i)}^{1-p_i} P(S_i | c_i = 0, \mathbf{y})}{\underbrace{P(c_i = 1 | \mathbf{y}_i)}_p P(S_i | c_i = 1, \mathbf{y})}$$

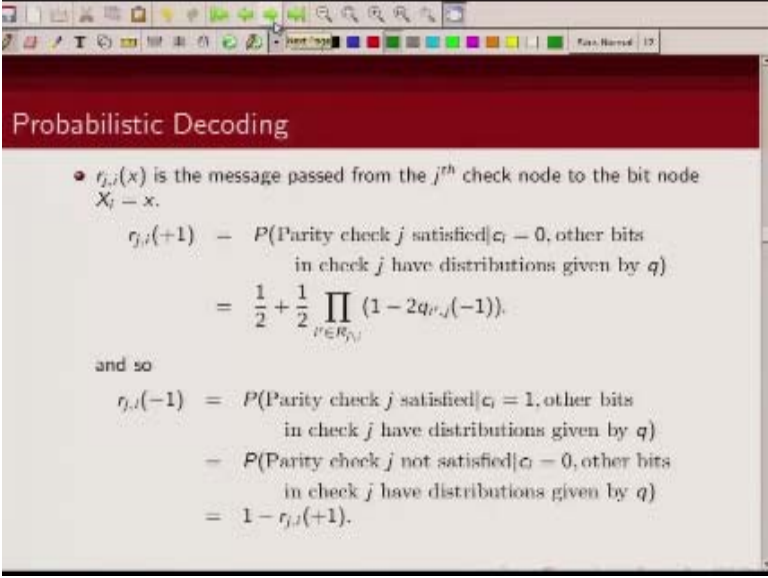
- Let's consider the term  $P(S_i | c_i = 0, \mathbf{y})$ . Given  $c_i = 0$ ,  $S_i$  holds if each of  $w_i$  parity checks involving  $c_i$  has the property that the  $w_i - 1$  bits in the check *other than*  $c_i$  have even parity.
- For parity check  $j \in C_i$ , the probability that the  $w_i - 1$  bits other than  $c_i$  have even parity is given by the lemma to be:

$$\frac{1}{2} + \frac{1}{2} \prod_{i' \in R_{i,j}} (1 - 2p_{i',j}(i)).$$

*RJV*

In my expression here this expression what I get is.

(Refer Slide Time: 23:17)



**Probabilistic Decoding**

- $r_{j,i}(x)$  is the message passed from the  $j^{\text{th}}$  check node to the bit node  $X_i = x$ .

$$r_{j,i}(+1) = P(\text{Parity check } j \text{ satisfied} | c_i = 0, \text{ other bits in check } j \text{ have distributions given by } q)$$

$$= \frac{1}{2} + \frac{1}{2} \prod_{i' \in R_{j,i}} (1 - 2q_{i',j}(-1)).$$

and so

$$r_{j,i}(-1) = P(\text{Parity check } j \text{ satisfied} | c_i = 1, \text{ other bits in check } j \text{ have distributions given by } q)$$

$$= P(\text{Parity check } j \text{ not satisfied} | c_i = 0, \text{ other bits in check } j \text{ have distributions given by } q)$$

$$= 1 - r_{j,i}(+1).$$

The expression for.

(Refer Slide Time: 23:19)

**Proof**

- From Bayes' rule:

$$\frac{P(c_i = 0 | \mathbf{y}, S_i)}{P(c_i = 1 | \mathbf{y}, S_i)} = \frac{\overbrace{P(c_i = 0 | y_i)}^{1-p_i} P(S_i | c_i = 0, \mathbf{y})}{\underbrace{P(c_i = 1 | y_i)}_{p_i} P(S_i | c_i = 1, \mathbf{y})}$$

- Let's consider the term  $P(S_i | c_i = 0, \mathbf{y})$ . Given  $c_i = 0$ ,  $S_i$  holds if each of  $w_i$  parity checks involving  $c_i$  has the property that the  $w_i - 1$  bits in the check *other than*  $c_i$  have even parity.
- For parity check  $j \in C_i$ , the probability that the  $w_i - 1$  bits other than  $c_i$  have even parity is given by the lemma to be:

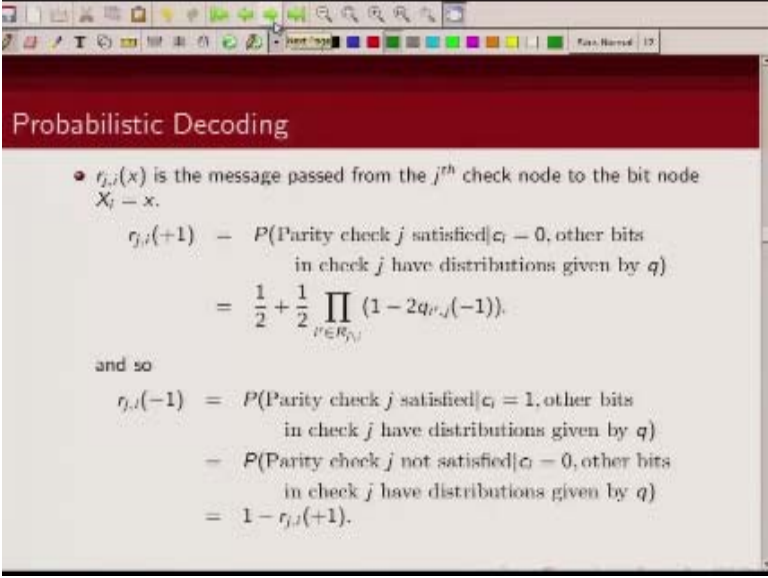
$$\frac{1}{2} + \frac{1}{2} \prod_{i' \in R_{j,i}} (1 - 2p_{i',j}(i)).$$

*R<sub>j,i</sub>*

This, okay.



(Refer Slide Time: 23:23)



**Probabilistic Decoding**

- $r_{j,i}(x)$  is the message passed from the  $j^{\text{th}}$  check node to the bit node  $X_i = x$ .

$$r_{j,i}(+1) = P(\text{Parity check } j \text{ satisfied} | c_i = 0, \text{ other bits in check } j \text{ have distributions given by } q)$$
$$= \frac{1}{2} + \frac{1}{2} \prod_{i' \in R_{j,i}} (1 - 2q_{i',j}(-1)).$$

and so

$$r_{j,i}(-1) = P(\text{Parity check } j \text{ satisfied} | c_i = 1, \text{ other bits in check } j \text{ have distributions given by } q)$$
$$= P(\text{Parity check } j \text{ not satisfied} | c_i = 0, \text{ other bits in check } j \text{ have distributions given by } q)$$
$$= 1 - r_{j,i}(+1).$$

So as we have said before we are writing, we are representing this LDPC code using this stenograph and there are 2 types of information which are getting propagated. One is one sort of messages which is from the message nodes going to the variable, going to the parity check nodes and this one set of message which from the parity check nodes coming back to the message nodes.

(Refer Slide Time: 23:49)

**Probabilistic Decoding**

- $r_{j,i}(x)$  is the message passed from the  $j^{\text{th}}$  check node to the bit node  $X_i = x$ .

$$r_{j,i}(+1) = P(\text{Parity check } j \text{ satisfied} | c_i = 0, \text{ other bits in check } j \text{ have distributions given by } q)$$

$$= \frac{1}{2} + \frac{1}{2} \prod_{i' \in R_{j,i}} (1 - 2q_{i',j}(-1)).$$

and so

$$r_{j,i}(-1) = P(\text{Parity check } j \text{ satisfied} | c_i = 1, \text{ other bits in check } j \text{ have distributions given by } q)$$

$$= P(\text{Parity check } j \text{ not satisfied} | c_i = 0, \text{ other bits in check } j \text{ have distributions given by } q)$$

$$= 1 - r_{j,i}(+1).$$


So we are denoting by  $R_{ji}$  the message which is passed from  $j$  parity check node to the  $i^{\text{th}}$  bit, we are denoting this by  $R_{ji}$  so what is  $r$ ,  $R_{ji}$  is the probability that  $j^{\text{th}}$  check node is satisfied given  $x$  is  $+1$ , so this is a probability that  $j^{\text{th}}$  check node is satisfied given  $C_i$  is 0 and other bits are given by distribution given by this  $q$ , now what is this  $q$  distribution?

(Refer Slide Time: 24:32)

### Probabilistic Decoding

- $q_{i,j}(x)$  is the message passed from the bit node  $X_i = x$  to the  $j^{\text{th}}$  check node.

$q_{i,j}(+1) = P(X_i = +1 | y_i, \text{ information from check nodes other than } j^{\text{th}} \text{ check node}).$

$$\frac{q_{i,j}(+1)}{q_{i,j}(-1)} = \frac{(1 - \rho_i) \prod_{j' \in C_{\setminus j}} r_{j',i}(+1)}{\rho_i \prod_{j' \in C_{\setminus j}} r_{j',i}(-1)}.$$


We will come to so that 2 type of messages, as I said there is one message so if I draw any tanner graph let us say just draw any tanner graph, let us say this some graph I am drawing, so there is one set that matches it which is going from message to this check nodes okay this is one set of messages which are going like this and there is another set of messages which is coming from the check node to the message bits. So we are denoting by  $q_i$  the message which is passed from bit node  $i$  to the  $j^{\text{th}}$  check node.

(Refer Slide Time: 25:12)

### Probabilistic Decoding

- $r_{j,i}(x)$  is the message passed from the  $j^{th}$  check node to the bit node  $X_i = x$ .

$$r_{j,i}(+1) = P(\text{Parity check } j \text{ satisfied} | c_i = 0, \text{ other bits in check } j \text{ have distributions given by } q)$$

$$= \frac{1}{2} + \frac{1}{2} \prod_{i' \in R_{j,i}} (1 - 2q_{i',j}(-1)).$$

and so

$$r_{j,i}(-1) = P(\text{Parity check } j \text{ satisfied} | c_i = 1, \text{ other bits in check } j \text{ have distributions given by } q)$$

$$= P(\text{Parity check } j \text{ not satisfied} | c_i = 0, \text{ other bits in check } j \text{ have distributions given by } q)$$

$$= 1 - r_{j,i}(+1).$$

And we are denoting by  $r_{j,i}$  the message which is passed from the check node to the bit node, now probability of this being.

(Refer Slide Time: 25:24)

### Probabilistic Decoding

- $r_{j,i}(x)$  is the message passed from the  $j^{\text{th}}$  check node to the bit node  $X_i = x$ .

$$r_{j,i}(+1) = P(\text{Parity check } j \text{ satisfied} | c_i = 0, \text{ other bits in check } j \text{ have distributions given by } q)$$

$$= \frac{1}{2} + \frac{1}{2} \prod_{i' \in R_j \setminus i} (1 - 2q_{i',j}(-1)).$$

and so

$$r_{j,i}(-1) = P(\text{Parity check } j \text{ satisfied} | c_i = 1, \text{ other bits in check } j \text{ have distributions given by } q)$$

$$= P(\text{Parity check } j \text{ not satisfied} | c_i = 0, \text{ other bits in check } j \text{ have distributions given by } q)$$

$$= 1 - r_{j,i}(+1).$$

X being +1 which is basically correspond to  $c_i$  their code bit being zero, this is this probability is defined as what is the probability that  $j^{\text{th}}$  parity check constrain is satisfied given that  $c_i$  the  $i^{\text{th}}$  bit is zero and other bits are given by distribution given by  $q_i$ . Now what is the probability that  $j^{\text{th}}$  parity check constrain will be satisfied given  $c_i$  will be zero, that probability is given by the condition that all other bits that are taking part in the parity check constrain other than  $c_i$  bit they should have even parity.

(Refer Slide Time: 26:09)

### Probabilistic Decoding

- $r_{j,i}(x)$  is the message passed from the  $j^{\text{th}}$  check node to the bit node  $X_i = x$ .

$$r_{j,i}(+1) = P(\text{Parity check } j \text{ satisfied} | c_i = 0, \text{ other bits in check } j \text{ have distributions given by } q)$$

$$= \frac{1}{2} + \frac{1}{2} \prod_{i' \in R_{j,i}} (1 - 2q_{i',j}(-1)).$$

and so

$$r_{j,i}(-1) = P(\text{Parity check } j \text{ satisfied} | c_i = 1, \text{ other bits in check } j \text{ have distributions given by } q)$$

$$= P(\text{Parity check } j \text{ not satisfied} | c_i = 0, \text{ other bits in check } j \text{ have distributions given by } q)$$

$$= 1 - r_{j,i}(+1).$$


And that is given by this expression, similarly we can find out what is the probability that  $r_{j,i}$  is - 1 this is 1 minus this probability.

(Refer Slide Time: 26:23)

### Probabilistic Decoding

•  $q_{i,j}(x)$  is the message passed from the bit node  $X_i = x$  to the  $j^{\text{th}}$  check node.

$q_{i,j}(+1) = P(X_i = +1 | y_i, \text{ information from check nodes other than } j^{\text{th}} \text{ check node}).$

$$\frac{q_{i,j}(+1)}{q_{i,j}(-1)} = \frac{(1 - p_i) \prod_{j' \in c_{i,j}} r_{j',i}(+1)}{p_i \prod_{j' \in c_{i,j}} r_{j',i}(-1)}.$$



Now what is this  $q_{i,j}$ ? It is a message passed from the bit notation  $i$  to the  $j^{\text{th}}$  parity check node.

(Refer Slide Time: 26:34)

### Probabilistic Decoding

- $q_{i,j}(x)$  is the message passed from the bit node  $X_i = x$  to the  $j^{\text{th}}$  check node.

$q_{i,j}(+1) = P(\underline{X_i = +1} | y_i, \text{ information from check nodes other than } j^{\text{th}} \text{ check node}).$

$$\frac{q_{i,j}(+1)}{q_{i,j}(-1)} = \frac{(1 - \rho_i)}{\rho_i} \frac{\prod_{j' \in C_{i,j}} r_{j',i}(+1)}{\prod_{j' \in C_{i,j}} r_{j',i}(-1)}$$


And this is so  $q_{i,j}$  here being +1 is given by what is the probability that  $x_i$  is +1 given receive sequence  $y_i$  and information from parity check nodes other than the  $j^{\text{th}}$  parity check node so, what is happening is when you are decoding because each bits are participating in multiple parity check equations, so you are getting some independent information from other parity check nodes and that information you want to pass it to and spread it around in this network.




(Refer Slide Time: 27:13)

### Probabilistic Decoding

$q_{i,j}(x)$  is the message passed from the bit node  $X_i = x$  to the  $j^{\text{th}}$  check node.

$q_{i,j}(+1) = P(\underline{X_i = +1} | y_i, \text{ information from check nodes other than } j^{\text{th}} \text{ check node}).$

$$\frac{q_{i,j}(+1)}{q_{i,j}(-1)} = \frac{(1 - p_i)}{p_i} \frac{\prod_{j' \in C_i \setminus j} r_{j',i}(+1)}{\prod_{j' \in C_i \setminus j} r_{j',i}(-1)}$$


And this probability is given by this expression.


(Refer Slide Time: 27:17)

### Probabilistic Decoding

- $q_{i,j}(x)$  is the message passed from the bit node  $X_i = x$  to the  $j^{\text{th}}$  check node.

$$q_{i,j}(+1) = P(X_i = +1 | y_i, \text{information from check nodes other than } j^{\text{th}} \text{ check node}).$$

$$\frac{q_{i,j}(+1)}{q_{i,j}(-1)} = \frac{(1 - \rho_i) \prod_{j' \in C_{i,j}} r_{j',i}(+1)}{\rho_i \prod_{j' \in C_{i,j}} r_{j',i}(-1)}$$



So there are two types of messages again I repeat which are being propagated in this graph, one is a message from the message nodes to the check nodes and then check nodes are sending some information saying okay whether these parity check constrain is satisfied or not, given input bit is one or zero and they are passing that information, so these information  $q_i$ 's are passed from message nodes to the check node and this message, these messages  $r_i$ 's have been passed from check nodes to the message node.

(Refer Slide Time: 27:54)

## Probabilistic Decoding

- $q_{i,j}(x)$  is the message passed from the bit node  $X_i = x$  to the  $j^{\text{th}}$  check node.

$$q_{i,j}(+1) = P(X_i = +1 | y_i, \text{ information from check nodes other than } j^{\text{th}} \text{ check node}).$$

$$\frac{q_{i,j}(+1)}{q_{i,j}(-1)} = \frac{(1 - p_i)}{p_i} \frac{\prod_{j' \in C_{i,j}} r_{j',i}(+1)}{\prod_{j' \in C_{i,j}} r_{j',i}(-1)}.$$



(Refer Slide Time: 27:55)

### Probabilistic Decoding

For all  $i, j$  such that  $h_{j,i} = 1$ . (So  $i$  indexes the bit nodes and  $j$  indexes the parity checks.)

- **Step 0:** Initialize:

So then how does this whole process go? So first step is

(Refer Slide Time: 28:00)

### Probabilistic Decoding

For all  $i, j$  such that  $h_{j,i} = 1$ . (So  $i$  indexes the bit nodes and  $j$  indexes the parity checks.)

- **Step 0:** Initialize:
  - Set  $p_i = P(c_i = 1 | Y_i = y_i) = 1 / (1 + \exp(2y_i / \sigma^2))$ .

We initialize the probabilities that we are going to send from the message nodes to the check node.

(Refer Slide Time: 28:08)

### Probabilistic Decoding

For all  $i, j$  such that  $h_{j,i} = 1$ . (So  $i$  indexes the bit nodes and  $j$  indexes the parity checks.)

- **Step 0:** Initialize:
  - Set  $p_i = P(c_i = 1 | Y_i = y_i) = 1 / (1 + \exp(2y_i / \sigma^2))$ .
  - $q_{i,j}(+1) = 1 - p_i$ .

So we calculate this  $p_i$ .

(Refer Slide Time: 28:10)

### Probabilistic Decoding

For all  $i, j$  such that  $h_{j,i} = 1$ . (So  $i$  indexes the bit nodes and  $j$  indexes the parity checks.)

- **Step 0:** Initialize:
  - Set  $p_i = P(c_i = 1 | Y_i = y_i) = 1 / (1 + \exp(2y_i / \sigma^2))$ .
  - $q_{i,j}(+1) = 1 - p_i$ .

And we calculate this  $q_i$ 's, these are messages we will send from the bit node to the parity check node okay.

(Refer Slide Time: 28:20)

## Probabilistic Decoding

For all  $i, j$  such that  $h_{j,i} = 1$ . (So  $i$  indexes the bit nodes and  $j$  indexes the parity checks.)

- **Step 0:** Initialize:

- Set  $p_i = P(c_i = 1 | Y_i = y_i) = 1 / (1 + \exp(2y_i / \sigma^2))$ .
- $q_{i,j}(+1) = 1 - p_i$ .
- $q_{i,j}(-1) = p_i$ .



(Refer Slide Time: 28:23)

### Probabilistic Decoding

For all  $i, j$  such that  $h_{j,i} = 1$ . (So  $i$  indexes the bit nodes and  $j$  indexes the parity checks.)

- **Step 0:** Initialize:
  - Set  $p_i = P(c_i = 1 | Y_i = y_i) = 1 / (1 + \exp(2y_i / \sigma^2))$ .
  - $q_{i,j}(+1) = 1 - p_i$ .
  - $q_{i,j}(-1) = p_i$ .

So that is the first step initialization step that we are calculating the initial messages that the bit nodes will send to the check node and that is basically based on channel likelihood values, it is given by this expression.

(Refer Slide Time: 28:42)

## Probabilistic Decoding

For all  $i, j$  such that  $h_{j,i} = 1$ . (So  $i$  indexes the bit nodes and  $j$  indexes the parity checks.)

- **Step 0:** Initialize:
  - Set  $p_i = P(c_i = 1 | Y_i = y_i) = 1 / (1 + \exp(2y_i / \sigma^2))$ .
  - $q_{i,j}(+1) = 1 - p_i$ .
  - $q_{i,j}(-1) = p_i$ .
- **Step 1:** Pass information from check nodes to bit nodes:

Next is once these messages are sent to check nodes then check nodes will do local computation and it will send this  $r_j$ 's back to the bit node.

(Refer Slide Time: 28:53)

## Probabilistic Decoding

For all  $i, j$  such that  $h_{j,i} = 1$ . (So  $i$  indexes the bit nodes and  $j$  indexes the parity checks.)

- **Step 0:** Initialize:
  - Set  $p_i = P(c_i = 1 | Y_i = y_i) = 1 / (1 + \exp(2y_i/\sigma^2))$ .
  - $q_{i,j}(+1) = 1 - p_i$ .
  - $q_{i,j}(-1) = p_i$ .
- **Step 1:** Pass information from check nodes to bit nodes.

So pass information from check nodes to the bit nodes so check nodes are going to pass these information  $r_i$ 's

(Refer Slide Time: 28:59)

## Probabilistic Decoding

For all  $i, j$  such that  $h_{j,i} = 1$ . (So  $i$  indexes the bit nodes and  $j$  indexes the parity checks.)

- **Step 0:** Initialize:

- Set  $p_i = P(c_i = 1 | Y_i = y_i) = 1 / (1 + \exp(2y_i / \sigma^2))$ .
- $q_{i,j}(+1) = 1 - p_i$ .
- $q_{i,j}(-1) = p_i$ .

- **Step 1:** Pass information from check nodes to bit nodes:

- $r_{i,j}(+1)$   $= \frac{1}{2} + \frac{1}{2} \prod_{i' \in R_{j,i}} (1 - 2q_{i',j}(-1))$

Back to the bit nodes.

(Refer Slide Time: 29:02)

## Probabilistic Decoding

For all  $i, j$  such that  $h_{j,i} = 1$ . (So  $i$  indexes the bit nodes and  $j$  indexes the parity checks.)

- **Step 0:** Initialize:

- Set  $p_i = P(c_i = 1 | Y_i = y_i) = 1 / (1 + \exp(2y_i / \sigma^2))$ .
- $q_{i,j}(+1) = 1 - p_i$ .
- $q_{i,j}(-1) = p_i$ .

- **Step 1:** Pass information from check nodes to bit nodes:

- $r_{j,i}(+1) = \frac{1}{2} + \frac{1}{2} \prod_{i' \in \mathcal{N}_j \setminus i} (1 - 2q_{i',j}(-1))$
- $r_{j,i}(-1) = 1 - r_{j,i}(+1)$ .

(Refer Slide Time: 29:04)

## Probabilistic Decoding

For all  $i, j$  such that  $h_{j,i} = 1$ . (So  $i$  indexes the bit nodes and  $j$  indexes the parity checks.)

- **Step 0:** Initialize:

- Set  $p_i = P(c_i = 1 | Y_i = y_i) = 1 / (1 + \exp(2y_i / \sigma^2))$ .
- $q_{i,j}(+1) = 1 - p_i$ .
- $q_{i,j}(-1) = p_i$ .

- **Step 1:** Pass information from check nodes to bit nodes.

- $r_{j,i}(+1) = \frac{1}{2} + \frac{1}{2} \prod_{j' \in R_j \setminus i} (1 - 2q_{j',j}(-1))$
- $r_{j,i}(-1) = 1 - r_{j,i}(+1)$ .

And this is given by this expression.

(Refer Slide Time: 29:07)



Probabilistic Decoding

- **Step 2:** Pass information from bit nodes to check nodes:

Now once you get this updated  $r_i$ 's from various check nodes then this  $q_i$ 's are updated.

(Refer Slide Time: 29:15)

Probabilistic Decoding

- **Step 2:** Pass information from bit nodes to check nodes.



(Refer Slide Time: 29:17)

### Probabilistic Decoding

- **Step 2:** Pass information from bit nodes to check nodes:
  - $q_{i,j}(+1) = K_{i,j}(1 - p_i) \prod_{j' \in C_{i,j}} r_{j',i}(+1)$
  - $q_{i,j}(-1) = K_{i,j}p_i \prod_{j' \in C_{i,j}} r_{j',i}(-1)$

So you are going to then send modified information, this  $q_i$ 's to the parity check nodes.

(Refer Slide Time: 29:25)

### Probabilistic Decoding

- **Step 2:** Pass information from bit nodes to check nodes:
  - $q_{i,j}(+1) = K_{i,j}(1 - p_i) \prod_{j' \in c_{i,j}} r_{j',i}(+1)$
  - $q_{i,j}(-1) = K_{i,j}p_i \prod_{j' \in c_{i,j}} r_{j',i}(-1)$

And this is basically given by this.

(Refer Slide Time: 29:28)

### Probabilistic Decoding

- **Step 2:** Pass information from bit nodes to check nodes:
  - $q_{i,j}(+1) = K_{i,j}(1 - p_i) \prod_{j' \in C_{i,j}} r_{j',i}(+1)$
  - $q_{i,j}(-1) = K_{i,j}p_i \prod_{j' \in C_{i,j}} r_{j',i}(-1)$
  - Here, the constants  $K_{i,j}$  are chosen so as to guarantee that  $q_{i,j}(+1) + q_{i,j}(-1) = 1$ .

You can do some normalization, these are  $K_i$ 's are just some constant, you can do some normalization.

(Refer Slide Time: 29:34)

### Probabilistic Decoding

- **Step 2:** Pass information from bit nodes to check nodes:
  - $q_{i,j}(+1) = K_{i,j}(1 - p_i) \prod_{j' \in C_{i,j}} r_{j',i}(+1)$
  - $q_{i,j}(-1) = K_{i,j}p_i \prod_{j' \in C_{i,j}} r_{j',i}(-1)$
  - Here, the constants  $K_{i,j}$  are chosen so as to guarantee that  $q_{i,j}(+1) + q_{i,j}(-1) = 1$ .
- **Step 3:** Compute the APP ratios for each bit position  $i$ :

And finally we will compute the A posteriori probability.

(Refer Slide Time: 29:40)

## Probabilistic Decoding

- **Step 2:** Pass information from bit nodes to check nodes:
  - $q_{i,j}(+1) = K_{i,j}(1 - p_i) \prod_{j' \in C_{i,j}} r_{j',i}(+1)$
  - $q_{i,j}(-1) = K_{i,j}p_i \prod_{j' \in C_{i,j}} r_{j',i}(-1)$
  - Here, the constants  $K_{i,j}$  are chosen so as to guarantee that  $q_{i,j}(+1) + q_{i,j}(-1) = 1$ .
- **Step 3:** Compute the APP ratios for each bit position  $i$ :
  - $Q_i(+1) = K_i(1 - p_i) \prod_{j \in C_i} r_{i,j}(+1)$

(Refer Slide Time: 29:41)

### Probabilistic Decoding

- **Step 2:** Pass information from bit nodes to check nodes:
  - $q_{i,j}(+1) = K_{i,j}(1 - p_i) \prod_{j' \in C_{i,j}} r_{j',i}(+1)$
  - $q_{i,j}(-1) = K_{i,j}p_i \prod_{j' \in C_{i,j}} r_{j',i}(-1)$
  - Here, the constants  $K_{i,j}$  are chosen so as to guarantee that  $q_{i,j}(+1) + q_{i,j}(-1) = 1$ .
- **Step 3:** Compute the APP ratios for each bit position  $i$ :
  - $Q_i(+1) = K_i(1 - p_i) \prod_{j \in C_i} r_{j,i}(+1)$
  - $Q_i(-1) = K_i p_i \prod_{j \in C_i} r_{j,i}(-1)$

And that is given by this expression.

(Refer Slide Time: 29:43)

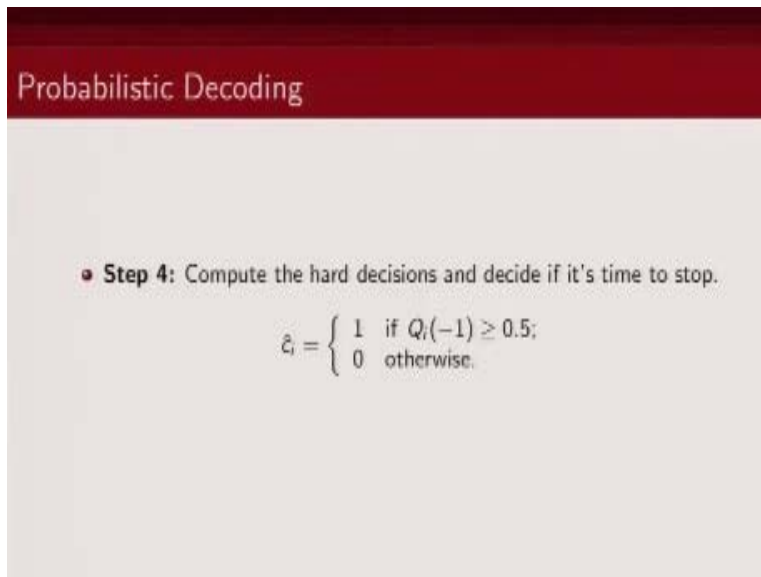
### Probabilistic Decoding

- **Step 2:** Pass information from bit nodes to check nodes:
  - $q_{i,j}(+1) = K_{i,j}(1 - p_i) \prod_{j' \in C_{i,j}} r_{j',i}(+1)$
  - $q_{i,j}(-1) = K_{i,j}p_i \prod_{j' \in C_{i,j}} r_{j',i}(-1)$
  - Here, the constants  $K_{i,j}$  are chosen so as to guarantee that  $q_{i,j}(+1) + q_{i,j}(-1) = 1$ .
- **Step 3:** Compute the APP ratios for each bit position  $i$ :
  - $Q_i(+1) = K_i(1 - p_i) \prod_{j \in C_i} r_{j,i}(+1)$
  - $Q_i(-1) = K_i p_i \prod_{j \in C_i} r_{j,i}(-1)$

And this we have derived earlier in the lecture okay. So to repeat basically how this whole process is going, you have this received values from the channel, based on that you compute your initial  $p_i$ 's and  $q_i$ 's, now this information are sent to a check nodes, now check nodes do local computation that what is the probability that the check node will be satisfied given a particular bit  $c_i$  is zero or one and then they pass that information to along the edges back to the bit nodes.

Now these bit nodes are getting information from other check nodes as well because each bit node is connected to multiple parity check equations. So it takes those input into a count to updates its  $q_i$ 's and this process is continued in an iterative fashion until all the parity check constraints are satisfied.

(Refer Slide Time: 30:43)



Probabilistic Decoding

- **Step 4:** Compute the hard decisions and decide if it's time to stop.

$$\hat{c}_i = \begin{cases} 1 & \text{if } Q_i(-1) \geq 0.5; \\ 0 & \text{otherwise.} \end{cases}$$

So finally basically once you compute the a postory probability then if  $q_i$  being -1 is more than 1 you decide in favor of 1 otherwise you decide in favor of zero.



(Refer Slide Time: 30:54)

Example

$c_0$	$c_1$	$c_2$
$c_3$	$c_4$	$c_5$
$c_6$	$c_7$	

• Consider the code with parity check matrix,  $\mathbf{H}$ :

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

So let us take an example to illustrate the decoding algorithm, so we have a low density parity check matrix given by this we have eight coded bits.

(Refer Slide Time: 31:07)

### Example

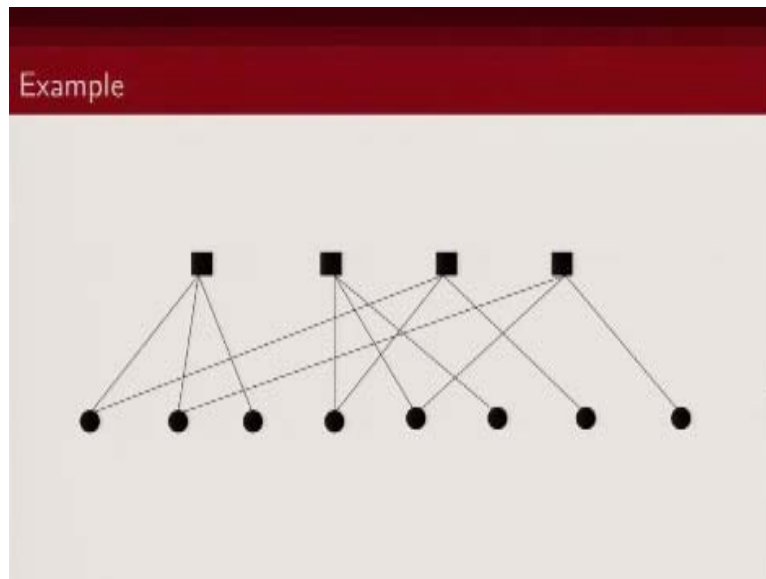
$c_0$	$c_1$	$c_2$
$c_3$	$c_4$	$c_5$
$c_6$	$c_7$	

- Consider the code with parity check matrix,  $\mathbf{H}$ :

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

- $n = 8$ ,  $m = n - k = 4$ ,  $d_{\min} = 3$

(Refer Slide Time: 31:10)



This is the tanner graph corresponding to the parity check matrix, we can just quickly check it.

(Refer Slide Time: 31:15)

### Example

$c_0$	$c_1$	$c_2$
$c_3$	$c_4$	$c_5$
$c_6$	$c_7$	

- Consider the code with parity check matrix,  $\mathbf{H}$ :

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

- $n = 8$ ,  $m = n - k = 4$ ,  $d_{\min} = 3$

(Refer Slide Time: 31:17)

Example

$c_0$	$c_1$	$c_2$
$c_3$	$c_4$	$c_5$
$c_6$	$c_7$	

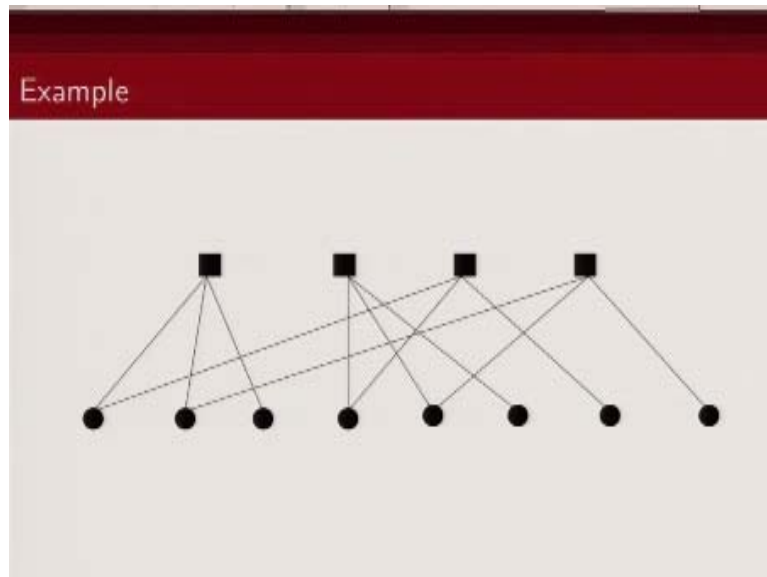
• Consider the code with parity check matrix,  $\mathbf{H}$ :

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

•  $n = 8, m = n - k = 4, d_{\min} = 3$

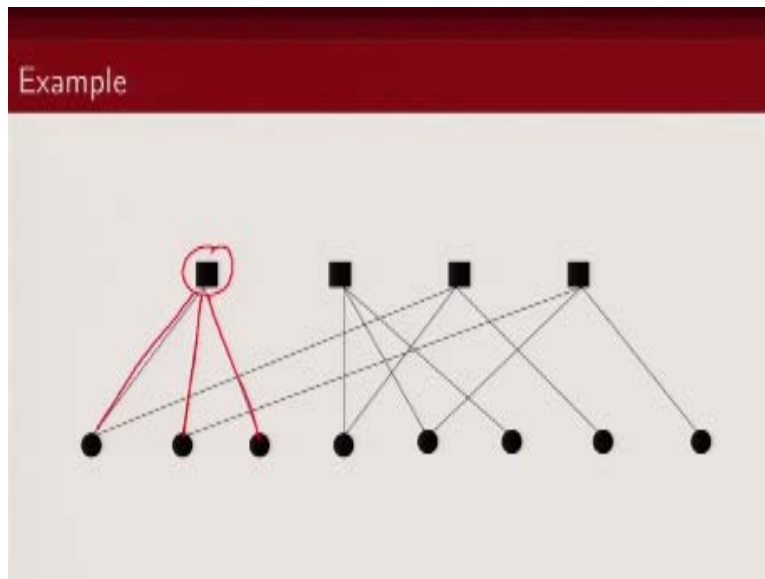
So the first parity check equation involves these three bits first three bits.

(Refer Slide Time: 31:22)



So you can see the first parity check equation involves this, this, and this.

(Refer Slide Time: 31:25)



(Refer Slide Time: 31:29)

Example

$c_0$	$c_1$	$c_2$
$c_3$	$c_4$	$c_5$
$c_6$	$c_7$	

• Consider the code with parity check matrix,  $\mathbf{H}$ :

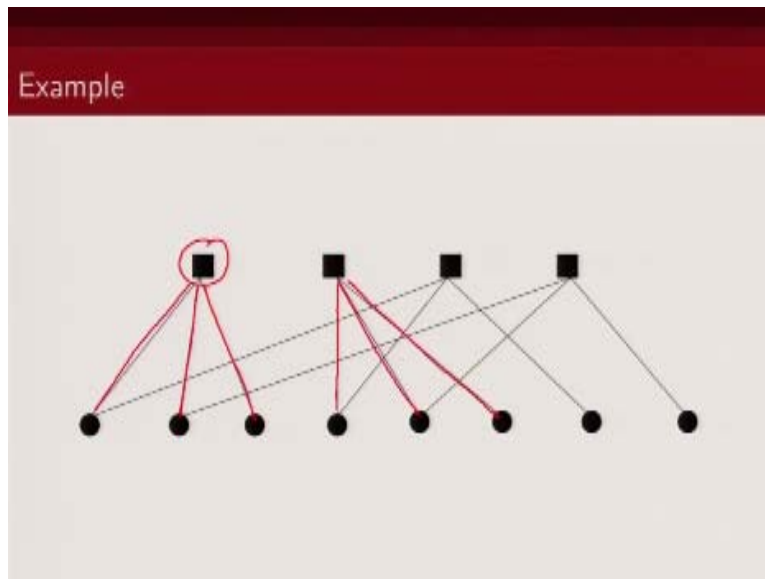
$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

•  $n = 8, m = n - k = 4, d_{\min} = 3$

Second parity check involves fourth, fifth, and sixth bit.



(Refer Slide Time: 31:34)



That is second parity check involves fourth, fifth, and sixth bit.

(Refer Slide Time: 31:41)

Example

$c_0$	$c_1$	$c_2$
$c_3$	$c_4$	$c_5$
$c_6$	$c_7$	

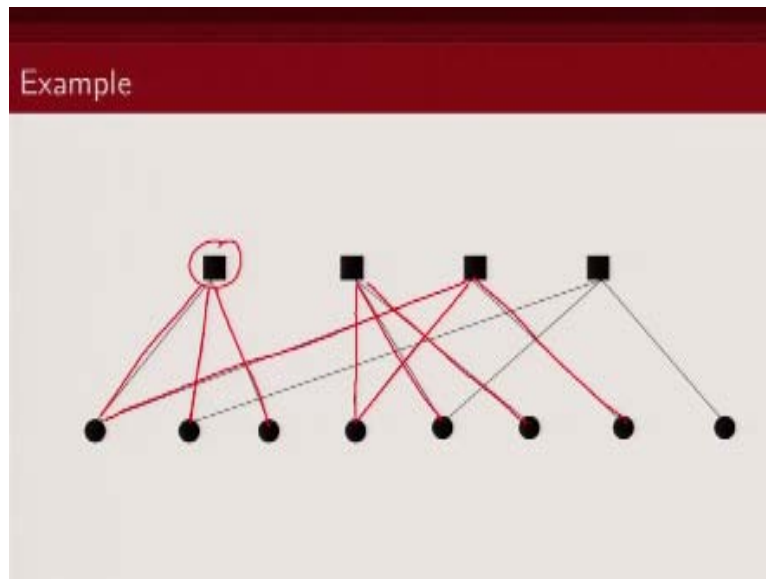
• Consider the code with parity check matrix,  $\mathbf{H}$ :

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

•  $n = 8, m = n - k = 4, d_{\min} = 3$

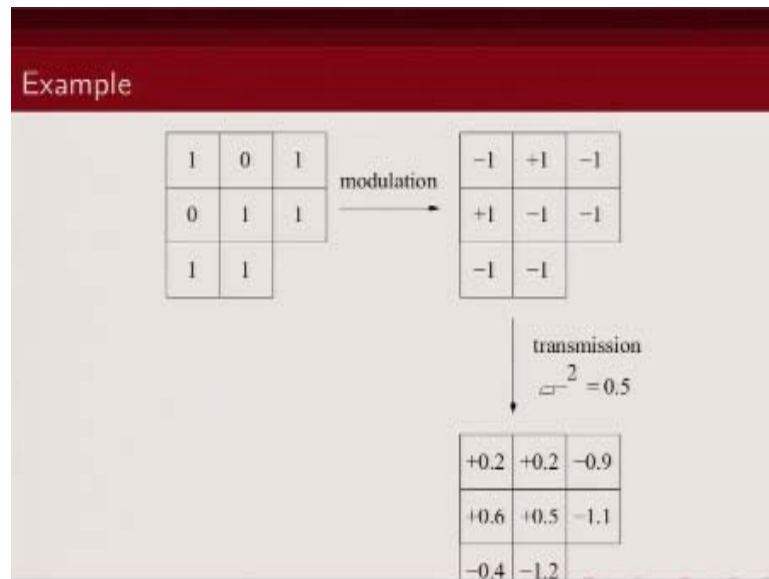
This involves first, fourth, seventh.

(Refer Slide Time: 31:48)



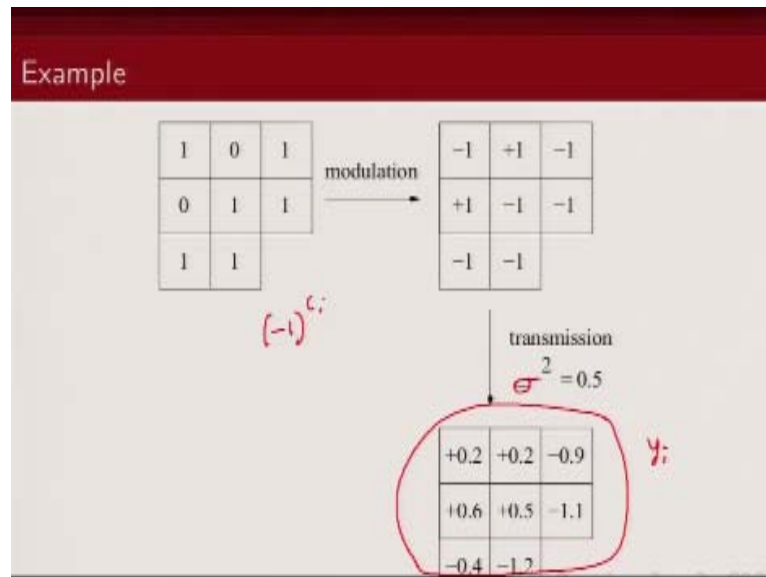
And that is first, fourth, and seventh, and similarly we can verify for this also okay, so this is a tanner graph corresponding to the parity check matrix that we have drawn.

(Refer Slide Time: 32:05)



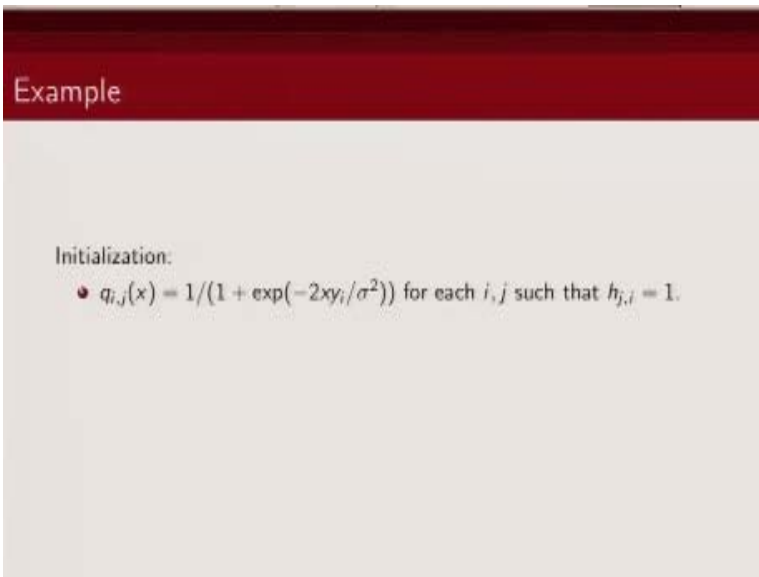
So these are the bits, eight bits which were transmitted now after modulation because we were mapping them as  $(-1)^{c_i}$ .

(Refer Slide Time: 32:14)



So one is getting mapped to -1 and zero is getting mapped to +1, so that is, these are the modulated bits and noise variance was 0.5 so what we get is these are the received value so these are my  $y_i$ 's okay. Now what is the first step? I will take this  $y_i$  and I will compute my  $p_i$ 's and  $q_i$ 's

(Refer Slide Time: 32:43)



Example

Initialization:

- $q_{i,j}(x) = 1 / (1 + \exp(-2xy_i/\sigma^2))$  for each  $i, j$  such that  $h_{j,i} = 1$ .

Initial  $p_i$  so I am going to first compute my initial  $q_i$ 's

(Refer Slide Time: 32:46)

## Example

Initialization:

- $q_{i,j}(x)$  =  $1/(1 + \exp(-2xy_i/\sigma^2))$  for each  $i, j$  such that  $h_{j,i} = 1$ .

(Refer Slide Time: 32:52)

### Example

Initialization:

- $q_{i,j}(x) = 1/(1 + \exp(-2xy_i/\sigma^2))$  for each  $i, j$  such that  $h_{j,i} = 1$ .
- $q_{0,0}(-1) = q_{0,2}(-1) = 0.310$  and  $q_{0,0}(+1) = q_{0,2}(-1) = 0.690$ .

And that is basically given by this expression.



(Refer Slide Time: 32:54)

## Example

Initialization:

- $q_{i,j}(x) = 1 / (1 + \exp(-2xy_i/\sigma^2))$  for each  $i, j$  such that  $h_{j,i} = 1$ .
- $q_{0,0}(-1) = q_{0,2}(-1) = 0.310$  and  $q_{0,0}(+1) = q_{0,2}(-1) = 0.690$ .
- $q_{1,0}(-1) = q_{1,3}(-1) = 0.310$  and  $q_{1,0}(+1) = q_{1,3}(+1) = 0.690$ .
- $q_{2,0}(-1) = 0.973$  and  $q_{2,0}(+1) = 0.027$ .


So I am just stating this, so I compute this  $q_i$ 's, now again just pay little attention to what  $q_i$ 's are.

(Refer Slide Time: 33:06)

### Probabilistic Decoding

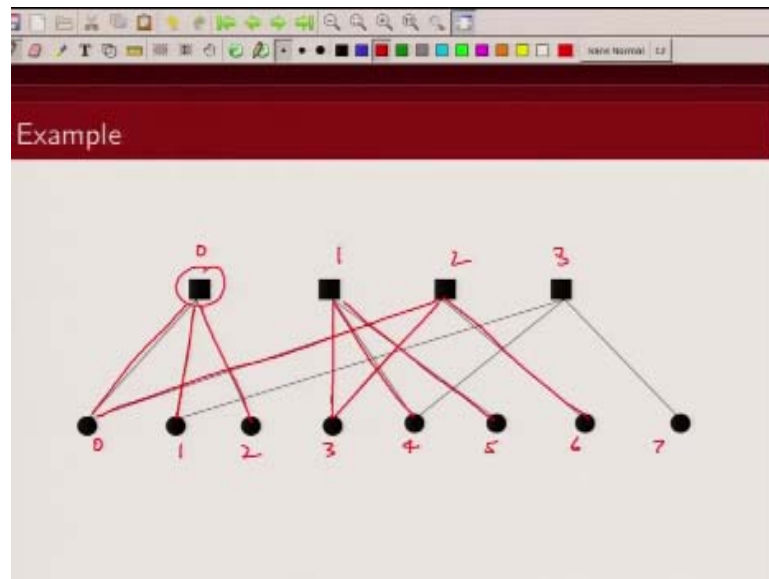
- $q_{i,j}(x)$  is the message passed from the bit node  $X_i = x$  to the  $j^{\text{th}}$  check node.

$q_{i,j}(+1) = P(X_i = +1 | y_i, \text{ information from check nodes other than } j^{\text{th}} \text{ check node}).$

$$\frac{q_{i,j}(+1)}{q_{i,j}(-1)} = \frac{(1 - p_i) \prod_{j' \in C_i \setminus j} r_{j',i}(+1)}{p_i \prod_{j' \in C_i \setminus j} r_{j',i}(-1)}.$$


$q_i$ 's are message passed from  $i^{\text{th}}$  bit to the  $j^{\text{th}}$  parity check constraint. So if you label your nodes from 0,1,2,3,4 as similarly label parity check constraints, so what you are going to notice is so you need to compute your  $q_i$ 's for, for in this example.

(Refer Slide Time: 33:33)




So this is 0, 1, 2, 3, 4, 5, 6, 7, similarly it is a 0, 1, 2, 3, so you will compute  $q_{0,0}$ .

(Refer Slide Time: 33:48)

**Probabilistic Decoding**

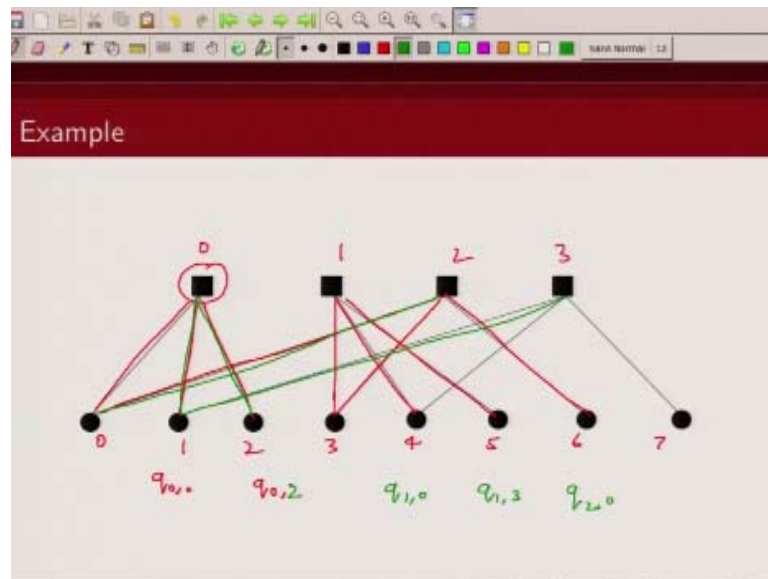
•  $q_{i,j}(x)$  is the message passed from the bit node  $X_i = x$  to the  $j^{\text{th}}$  check node.

$q_{i,j}(+1) = P(X_i = +1 | y_i, \text{ information from check nodes other than } j^{\text{th}} \text{ check node}).$

$$\frac{q_{i,j}(+1)}{q_{i,j}(-1)} = \frac{(1 - p_i) \prod_{j' \in C_i \setminus j} r_{j',i}(+1)}{p_i \prod_{j' \in C_i \setminus j} r_{j',i}(-1)}.$$


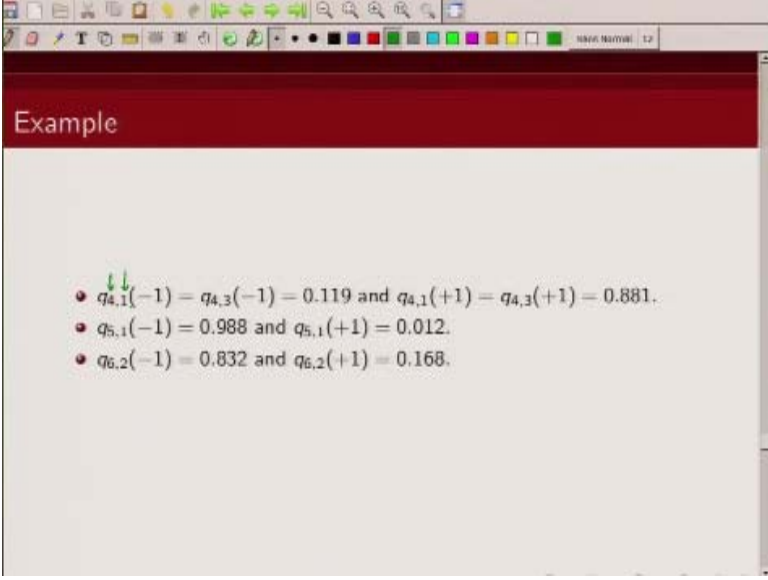
So  $q_{i,j}$  is the message passed from  $i^{\text{th}}$  node to the  $j^{\text{th}}$  check node right. So  $q_{ij}$ 's you need to compute for in this particular example.

(Refer Slide Time: 34:02)



So let us look at 0<sup>th</sup> node, so you need to compute  $q_{00}$  because your message you are sending from 0<sup>th</sup> node to 0<sup>th</sup> parity check. You need to compute  $q_{02}$  because you are sending message from 0<sup>th</sup> bit to the second this thing. You need to compute this  $q_{10}$ , you need to compute  $q_{13}$  right. You need to compute  $q_{20}$  and how are these initially computed, these are initially computed based on what are the received values  $y_i$ 's that we have received. And that is precisely what I have shown here.

(Refer Slide Time: 35:02)

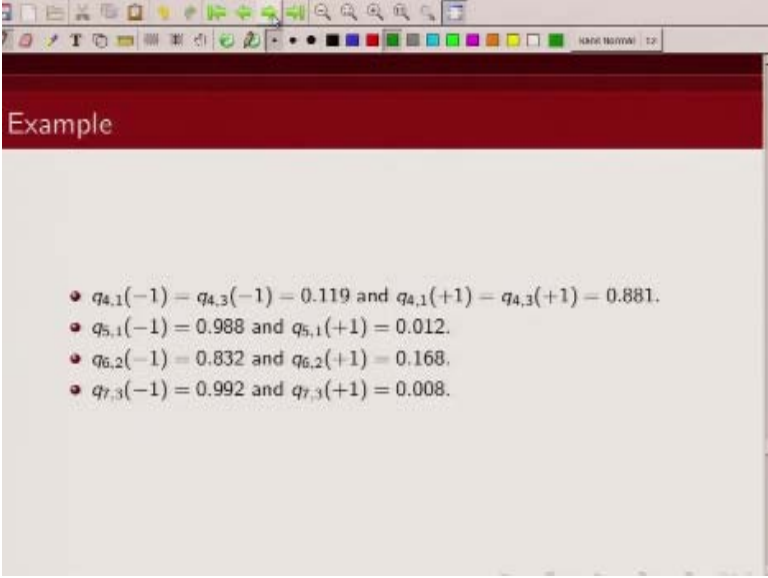


The slide is titled "Example" and contains three bullet points, each with a red circular icon. The first bullet point has two green arrows pointing down to the first two subscripts of the first expression. The expressions are as follows:

- $q_{4,1}(-1) = q_{4,3}(-1) = 0.119$  and  $q_{4,1}(+1) = q_{4,3}(+1) = 0.881$ .
- $q_{5,1}(-1) = 0.988$  and  $q_{5,1}(+1) = 0.012$ .
- $q_{6,2}(-1) = 0.832$  and  $q_{6,2}(+1) = 0.168$ .

So I have, you can see here these are the computed values of  $q_i$ 's and pay close attention to these instances, these denote that  $i^{\text{th}}$  bit is sending information to the  $j^{\text{th}}$  parity check constraint okay.

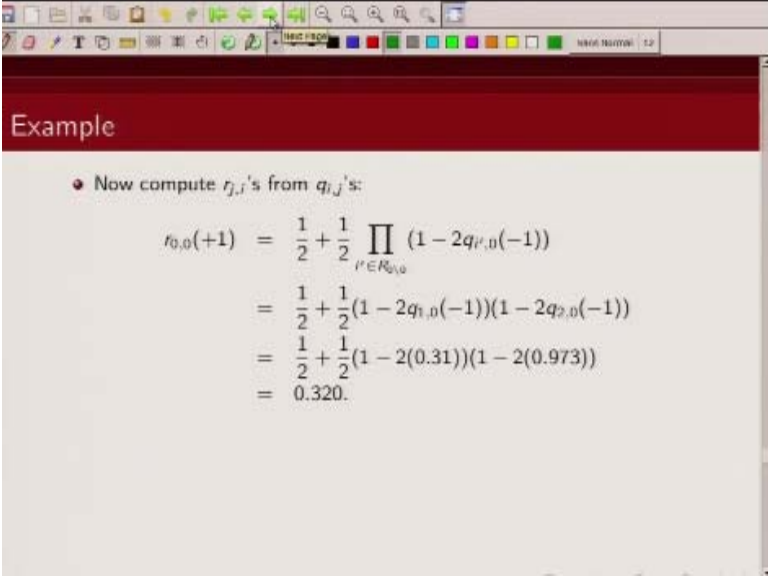
(Refer Slide Time: 35:24)



The image shows a presentation slide with a dark red header bar containing the word "Example" in white. Below the header, on a light gray background, there is a bulleted list of four mathematical equations. Each equation is preceded by a small red circular icon. The equations relate values of  $q_{i,j}$  at  $-1$  and  $+1$  for different pairs of  $i$  and  $j$ .

- $q_{4,1}(-1) = q_{4,3}(-1) = 0.119$  and  $q_{4,1}(+1) = q_{4,3}(+1) = 0.881.$
- $q_{5,1}(-1) = 0.988$  and  $q_{5,1}(+1) = 0.012.$
- $q_{6,2}(-1) = 0.832$  and  $q_{6,2}(+1) = 0.168.$
- $q_{7,3}(-1) = 0.992$  and  $q_{7,3}(+1) = 0.008.$

(Refer Slide Time: 35:25)



Example

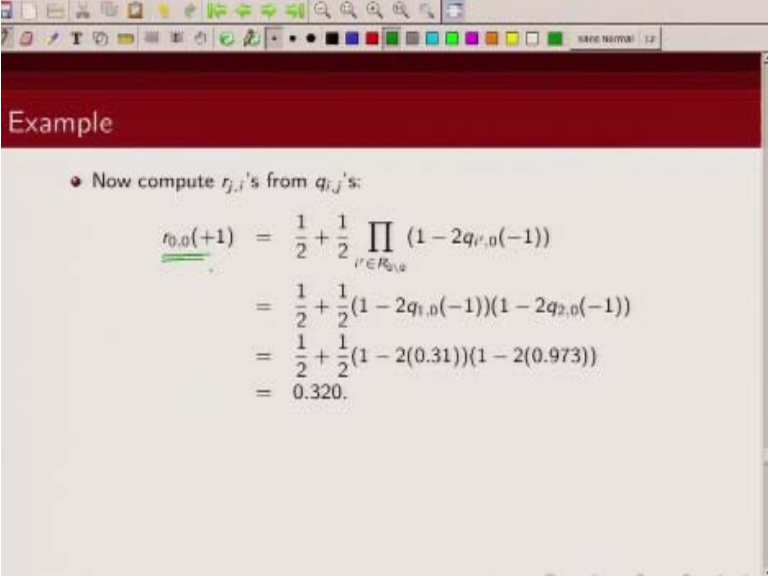
- Now compute  $r_{j,i}$ 's from  $q_{i,j}$ 's:

$$\begin{aligned}r_{0,0}(+1) &= \frac{1}{2} + \frac{1}{2} \prod_{i' \in R_{0,0}} (1 - 2q_{i',0}(-1)) \\&= \frac{1}{2} + \frac{1}{2} (1 - 2q_{1,0}(-1))(1 - 2q_{2,0}(-1)) \\&= \frac{1}{2} + \frac{1}{2} (1 - 2(0.31))(1 - 2(0.973)) \\&= 0.320.\end{aligned}$$

So once we have computed this  $q_{ij}$  the next thing that we need to do is these parity check constraints are now going to check okay given that the bit is zero. What is the probability that parity checks constraint is satisfied, given the bit is one what is the probability that parity check constrain is satisfied?



(Refer Slide Time: 35:50)



Example

• Now compute  $r_{j,i}$ 's from  $q_{i,j}$ 's:

$$\begin{aligned} \underline{r_{0,0}(+1)} &= \frac{1}{2} + \frac{1}{2} \prod_{i' \in R_0 \setminus \{0\}} (1 - 2q_{i',0}(-1)) \\ &= \frac{1}{2} + \frac{1}{2} (1 - 2q_{1,0}(-1))(1 - 2q_{2,0}(-1)) \\ &= \frac{1}{2} + \frac{1}{2} (1 - 2(0.31))(1 - 2(0.973)) \\ &= 0.320. \end{aligned}$$

And those are given by these  $r$ 's,  $r_{i,j}$ 's right. What is the probability that  $0^{\text{th}}$  parity check constraint is satisfied given the  $0^{\text{th}}$  bit is +1. Now what is that probability, that probability is given by, so we are looking at first parity check constraint. Now given that first bits  $C_i$  is zero what we want is the other bits which are involved in the parity check constraints they should have even parity. So let us look at the H matrix for the first row.

(Refer Slide Time: 36:34)

Example

$c_0$	$c_1$	$c_2$
$c_3$	$c_4$	$c_5$
$c_6$	$c_7$	

• Consider the code with parity check matrix,  $H$ :

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \rightarrow R_0 = \{0, 1, 2\}$$

So this is  $R_0$ ,  $R_0$  is 0, 1, and 2. So given that this bit is zero what we want to find out it is what is the probability that sum of these two add up to even parity. And that is what we are doing here.

(Refer Slide Time: 36:58)

Example

• Now compute  $r_{j,i}$ 's from  $q_{i,j}$ 's:

$$r_{0,0}(+1) = \frac{1}{2} + \frac{1}{2} \prod_{i' \in R_0 \setminus 0} (1 - 2q_{i',0}(-1))$$

$$= \frac{1}{2} + \frac{1}{2} (1 - 2q_{1,0}(-1))(1 - 2q_{2,0}(-1))$$

$$= \frac{1}{2} + \frac{1}{2} (1 - 2(0.31))(1 - 2(0.973))$$

$$= 0.320.$$

$R_0 = \{0, 1, 2\}$   
 $R_0 \setminus 0 = \{1, 2\}$

Note here,  $i^{\text{th}}$  belongs to  $R_0$  minus this zero term. So here you will have two terms, one is corresponding to  $i$  being one and other  $i$  being two. Why because your  $R_0$  was 0, 1, 2, so  $R_0$  minus this element zero is nothing but 1 and 2. So here in this product you will have two terms, one corresponding to  $q_{1,0}$  other corresponding to  $q_{2,0}$ . Because in that  $0^{\text{th}}$  parity check equation other than the first bit, the bits that are participating is that other than

(Refer Slide Time: 37:44)

Example

• Now compute  $r_{j,i}$ 's from  $q_{i,j}$ 's:

$$\begin{aligned}
 \underline{r_{0,0}(+1)} &= \frac{1}{2} + \frac{1}{2} \prod_{i' \in R_{0 \setminus 0}} (1 - 2q_{i',0}(-1)) \\
 &= \frac{1}{2} + \frac{1}{2} (1 - 2q_{1,0}(-1))(1 - 2q_{2,0}(-1)) \\
 &= \frac{1}{2} + \frac{1}{2} (1 - 2(0.31))(1 - 2(0.973)) \\
 &= 0.320.
 \end{aligned}$$

$R_0 = \{0, 1, 2\}$   
 $R_0 \setminus 0 = \{1, 2\}$

The 0<sup>th</sup> bit other bits which are participating is bit number one and bit number two. So this is the probability that parity check constraint is satisfied given  $C_i$  is zero.

(Refer Slide Time: 37:57)

**Example**

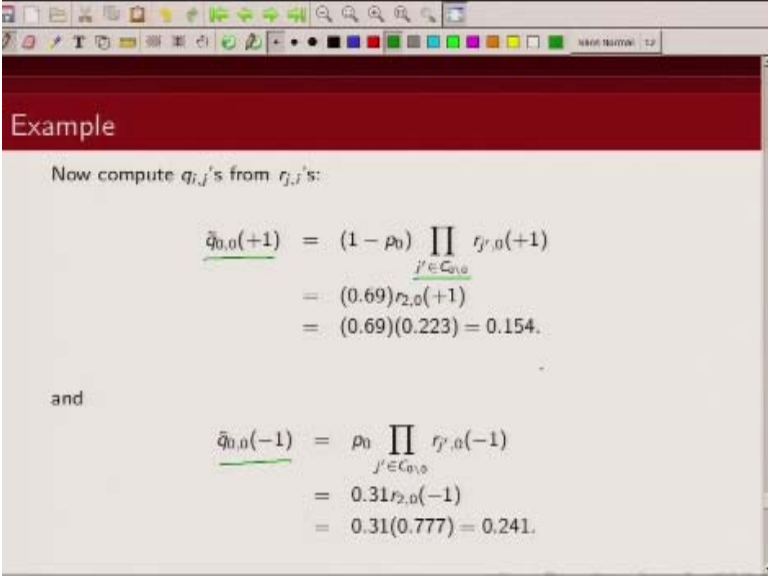
- Now compute  $r_{j,i}$ 's from  $q_{i,j}$ 's:

$$\begin{aligned}
 r_{0,0}(+1) &= \frac{1}{2} + \frac{1}{2} \prod_{i' \in R_{0,0}} (1 - 2q_{i',0}(-1)) \\
 &= \frac{1}{2} + \frac{1}{2} (1 - 2q_{1,0}(-1))(1 - 2q_{2,0}(-1)) \\
 &= \frac{1}{2} + \frac{1}{2} (1 - 2(0.31))(1 - 2(0.973)) \\
 &= 0.320.
 \end{aligned}$$

- In a similar way:
  - $r_{0,1}(+1) = 0.5 + 0.5(1 - 2(0.31))(1 - 2(0.973)) = 0.32$
  - $r_{0,2}(+1) = 0.5 + 0.5(1 - 2(0.31))(1 - 2(0.31)) = 0.57$
  - $r_{1,1}(+1) = 0.5 + 0.5(1 - 2(0.119))(1 - 2(0.988)) = 0.128$
  - $r_{2,0}(+1) = 0.5 + 0.5(1 - 2(0.083))(1 - 2(0.832)) = 0.223.$
- $r_{j,i}(-1) = 1 - r_{j,i}(+1).$

And similarly we can calculate the other  $r_i$ 's. I am not going into detail of that it is just the same procedure repeated. So once we calculate this  $q_i$ 's,  $r_i$ 's and initial  $q_i$ 's.

(Refer Slide Time: 38:12)



Example

Now compute  $q_{i,j}$ 's from  $r_{j,j}$ 's:

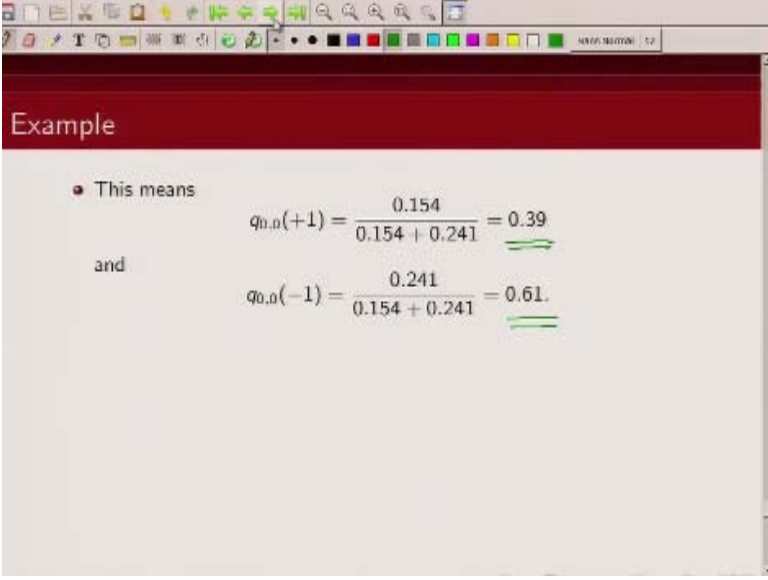
$$\begin{aligned}\bar{q}_{0,0}(+1) &= (1 - \rho_0) \prod_{j' \in C_{0,0}} r_{j',0}(+1) \\ &= (0.69)r_{2,0}(+1) \\ &= (0.69)(0.223) = 0.154.\end{aligned}$$

and

$$\begin{aligned}\bar{q}_{0,0}(-1) &= \rho_0 \prod_{j' \in C_{0,0}} r_{j',0}(-1) \\ &= 0.31r_{2,0}(-1) \\ &= 0.31(0.777) = 0.241.\end{aligned}$$

Then we are going to update our  $q_i$ 's. And again we follow, so we find the product over all those check equations other than that particular bit and we repeat this for  $q_{i,j}$  being +1 and -1. And we can normalize these probabilities so that they sum up to.

(Refer Slide Time: 38:43)



The image shows a presentation slide with a red header bar containing the word "Example". Below the header, the text "• This means" is followed by the equation  $q_{0,0}(+1) = \frac{0.154}{0.154 + 0.241} = 0.39$ . The denominator and the result are underlined with green lines. Below this, the word "and" is followed by the equation  $q_{0,0}(-1) = \frac{0.241}{0.154 + 0.241} = 0.61$ . The denominator and the result are also underlined with green lines.

Example

• This means

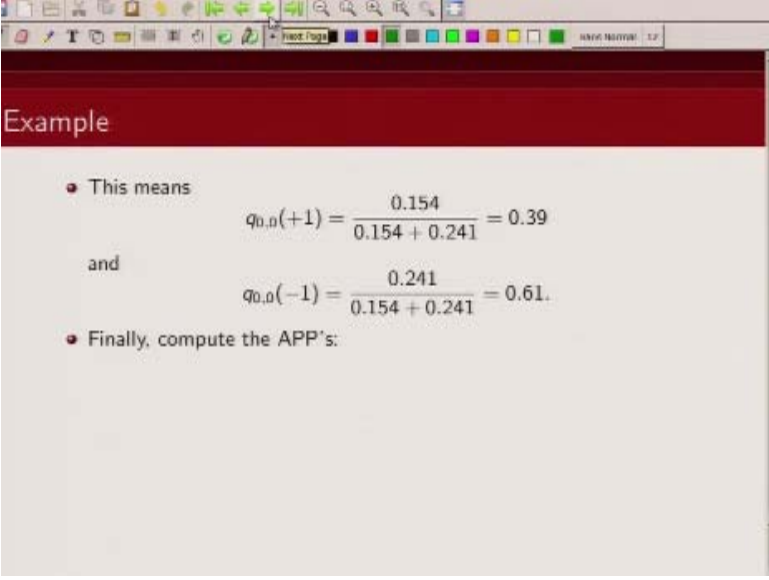
$$q_{0,0}(+1) = \frac{0.154}{0.154 + 0.241} = 0.39$$

and

$$q_{0,0}(-1) = \frac{0.241}{0.154 + 0.241} = 0.61$$

One, so if we normalize it these are the probabilities that we are getting right.

(Refer Slide Time: 38:50)



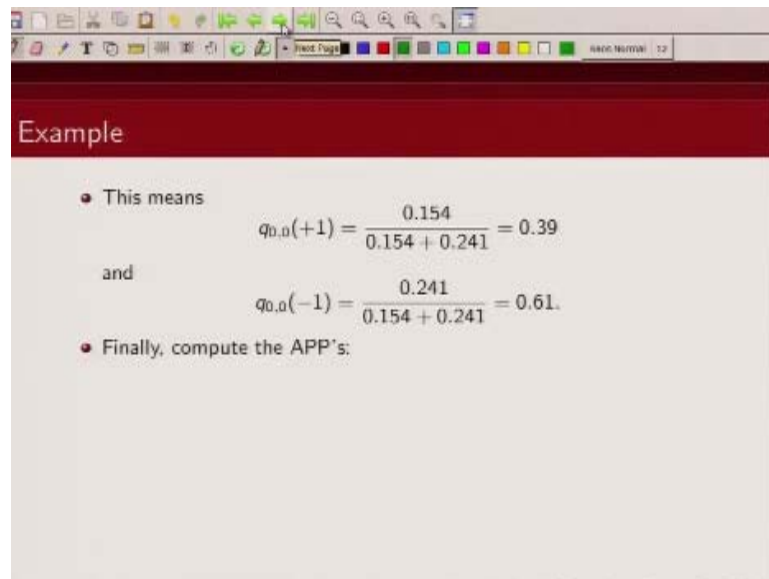
The slide is titled "Example" and contains the following content:

- This means
$$q_{0,0}(+1) = \frac{0.154}{0.154 + 0.241} = 0.39$$
- and
$$q_{0,0}(-1) = \frac{0.241}{0.154 + 0.241} = 0.61.$$
- Finally, compute the APP's:

And next thing finally after we have computed one round of iteration, what is one round of iteration, you send these  $q_i$ 's to a check node then check node gives you backs  $r_i$ 's you update your  $q_i$ 's.



(Refer Slide Time: 39:06)

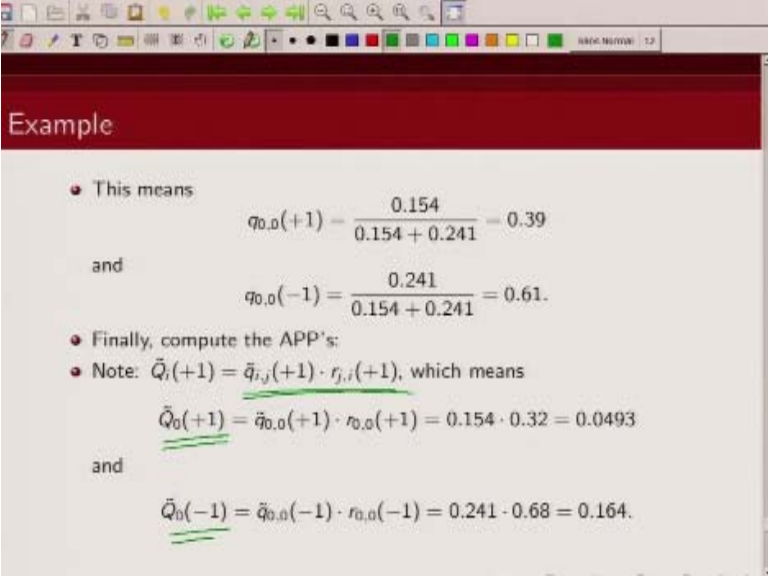


**Example**

- This means
$$q_{0,0}(+1) = \frac{0.154}{0.154 + 0.241} = 0.39$$
- and
$$q_{0,0}(-1) = \frac{0.241}{0.154 + 0.241} = 0.61$$
- Finally, compute the APP's:

And once you do that.

(Refer Slide Time: 39:08)



The slide is titled "Example" and contains the following content:

- This means
$$q_{0,0}(+1) = \frac{0.154}{0.154 + 0.241} = 0.39$$
- and
$$q_{0,0}(-1) = \frac{0.241}{0.154 + 0.241} = 0.61.$$
- Finally, compute the APP's:
- Note:  $\tilde{Q}_i(+1) = \tilde{q}_{i,j}(+1) \cdot r_{j,i}(+1)$ , which means
$$\tilde{Q}_0(+1) = \tilde{q}_{0,0}(+1) \cdot r_{0,0}(+1) = 0.154 \cdot 0.32 = 0.0493$$
- and
$$\tilde{Q}_0(-1) = \tilde{q}_{0,0}(-1) \cdot r_{0,0}(-1) = 0.241 \cdot 0.68 = 0.164.$$

You can then find out the A posteriori probability which we have done here. So probability of wit being one, probability of this being zero, and then based on again you can normalize.

(Refer Slide Time: 39:22)

Example

- This yields the APP

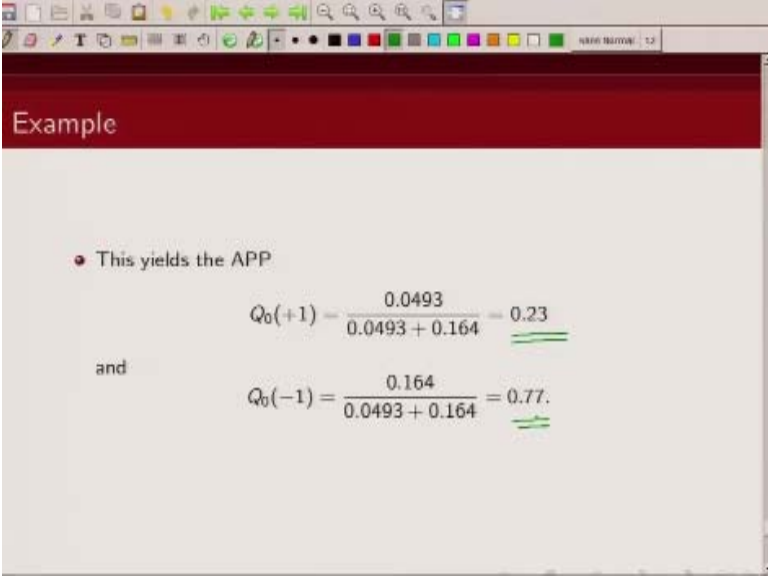
$$Q_0(+1) = \frac{0.0493}{0.0493 + 0.164} = \underline{\underline{0.23}}$$

and

$$Q_0(-1) = \frac{0.164}{0.0493 + 0.164} = \underline{\underline{0.77}}$$

These probabilities so that comes out to be this. Now based on whichever is more likely you decide in favor of that. In this case  $q$  being  $-1$  is more likely so a bit which was transmitted was one okay. So to recap how this probabilistic decoding algorithm works, you have your receive six sequence. You compute this probabilities  $q_i$ 's message bits, send it to the check bits, check bits then do some compute local computation sends the information back. Now this process goes on and on until basically all the parity check constraints are satisfied.

(Refer Slide Time: 40:07)



The screenshot shows a presentation slide with a dark red header containing the word "Example". Below the header, there is a bullet point that reads "This yields the APP". To the right of this text, two equations are displayed. The first equation is  $Q_0(+1) = \frac{0.0493}{0.0493 + 0.164} = \underline{\underline{0.23}}$ . The word "and" is placed to the left of the second equation, which is  $Q_0(-1) = \frac{0.164}{0.0493 + 0.164} = \underline{\underline{0.77}}$ . The slide is viewed through a software window with a standard toolbar at the top.

Example

- This yields the APP

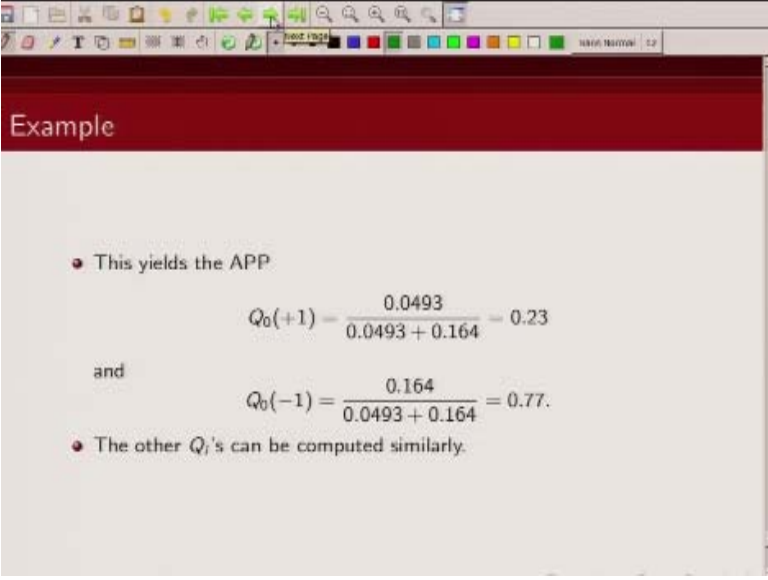
$$Q_0(+1) = \frac{0.0493}{0.0493 + 0.164} = \underline{\underline{0.23}}$$

and

$$Q_0(-1) = \frac{0.164}{0.0493 + 0.164} = \underline{\underline{0.77}}$$

Or the maximum number iterations have reached.

(Refer Slide Time: 40:09)



The slide is titled "Example" and contains the following content:

- This yields the APP

$$Q_0(+1) = \frac{0.0493}{0.0493 + 0.164} = 0.23$$

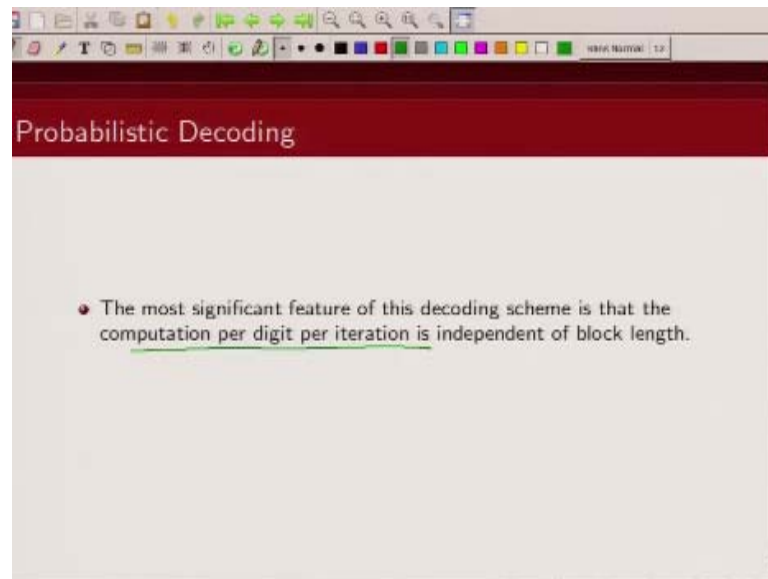
and

$$Q_0(-1) = \frac{0.164}{0.0493 + 0.164} = 0.77.$$

- The other  $Q_i$ 's can be computed similarly.

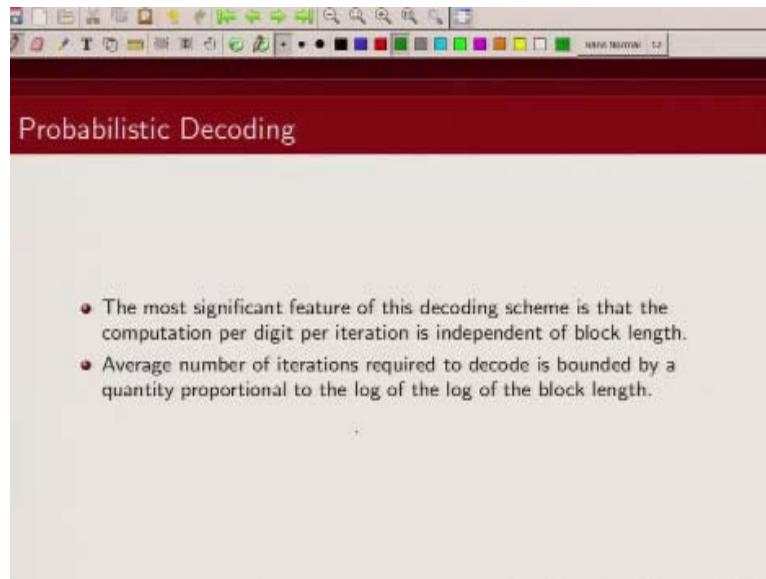
Now this we have done for the first bit you can do the same thing for other bits as well. In fact, the beauty of this algorithm is you can do this whole operation parallelly.

(Refer Slide Time: 40:22)



So one of the nicest feature of this is, this algorithm is computation per bit per digit is independent of block size.

(Refer Slide Time: 40:30)



And average number of iterations required basically your decode is typically bounded by  $\log, \log$  of  $n$ . After that you started getting code related information back. So with this we will conclude our discussion on probabilistic decoding of LDPC codes. Thank you.

### **Acknowledgement**

**Ministry of Human Resource & Development**

**Prof. Satyaki Roy**

**Co-ordinator, NPTEL IIT Kanpur**

**NPTEL Team**

**Sanjay Pal**

**Ashish Singh**

**Badal Pradhan**

**Tapobrata Das**

**Ram Chandra**

**Dilip Tripathi**

**Manoj Shrivastava**

**Padam Shukla**

**Sanjay Mishra**

**Shubham Rawat**

**Shikha Gupta**

**K. K. Mishra**

**Aradhana Singh**

**Sweta  
Ashutosh Gairola  
Dilip Katiyar  
Sharwan  
Hari Ram  
Bhadra Rao  
Puneet Kumar Bajpai  
Lalty Dutta  
Ajay Kanaujia  
Shivendra Kumar Tiwari**

**an IIT Kanpur Production**

**©copyright reserved**