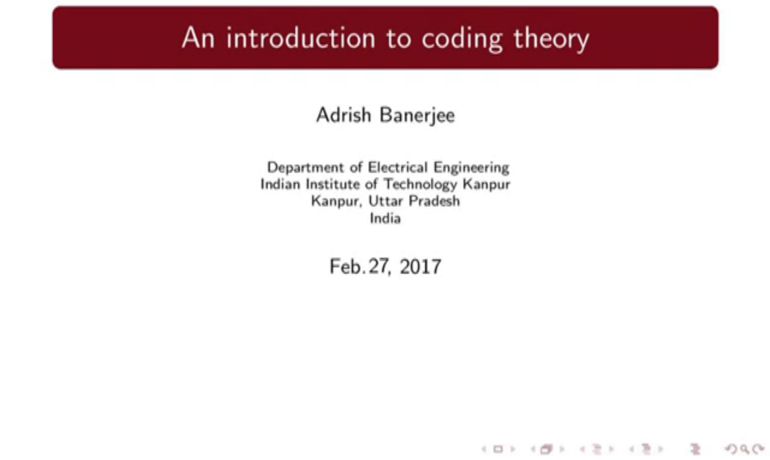
An Introduction to Coding Theory
Professor Adrish Banerji
Department of Electrical Engineering
Indian Institute of Technology, Kanpur
Module 06
Lecture Number 26
Decoding of low density parity check codes-II: Belief Propogation Algorithm

(Refer Slide Time 00:14)



An introduction to coding theory

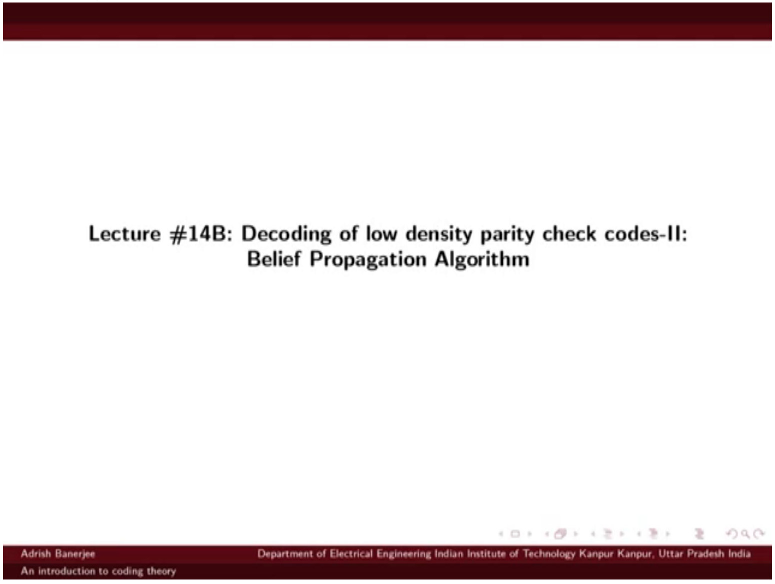
Adrish Banerjee

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

Feb.27, 2017

We will continue our discussion on decoding of

(Refer Slide Time 00:17)



Lecture #14B: Decoding of low density parity check codes-II:
Belief Propagation Algorithm

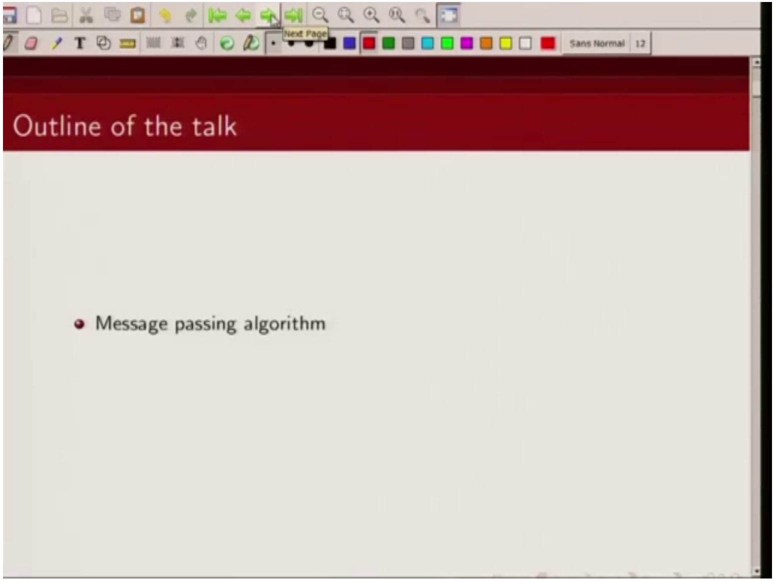
Adrish Banerjee Department of Electrical Engineering Indian Institute of Technology Kanpur Kanpur, Uttar Pradesh India
An introduction to coding theory

L D P C codes. Today we are going to talk about

(Refer Slide Time 00:21)

probabilistic decoding algorithm. So

(Refer Slide Time 00:24)

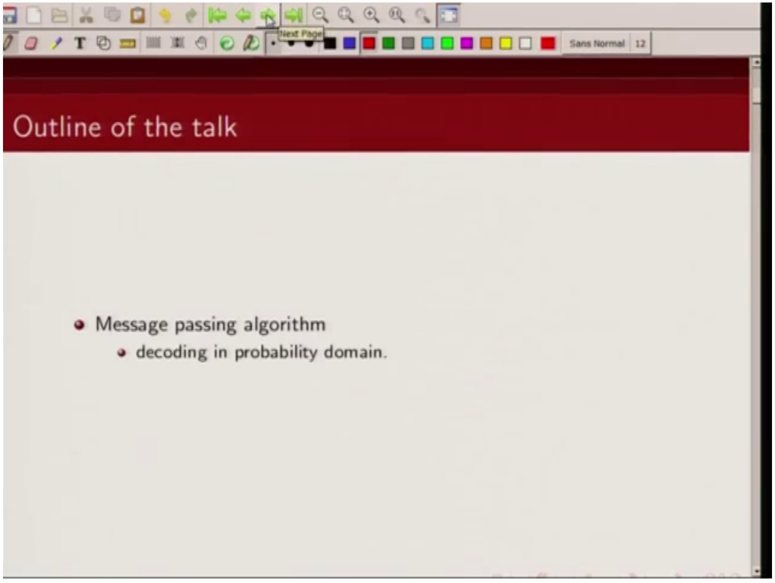


Outline of the talk

- Message passing algorithm

we are going to talk about belief propagation algorithm. So before we do that,

(Refer Slide Time 00:30)

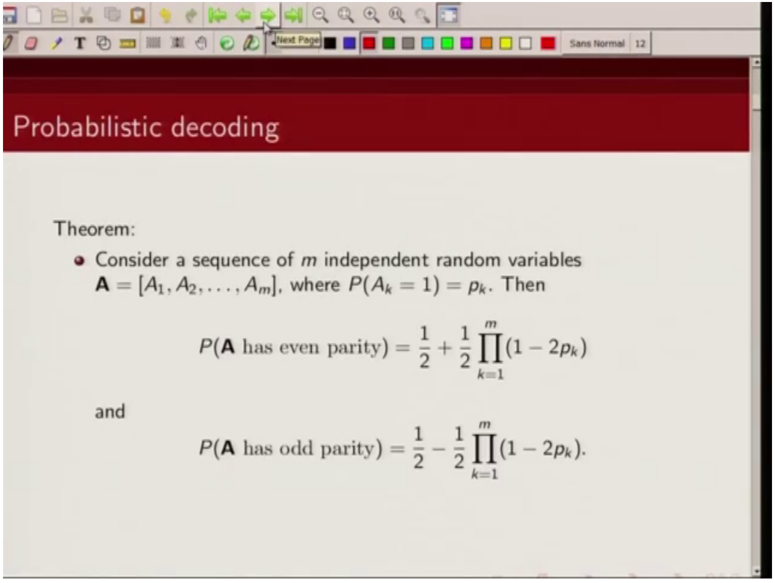


Outline of the talk

- Message passing algorithm
  - decoding in probability domain.

we are going to prove some results and use these results for decoding of L D P C codes.

(Refer Slide Time 00:38)



Probabilistic decoding

Theorem:

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

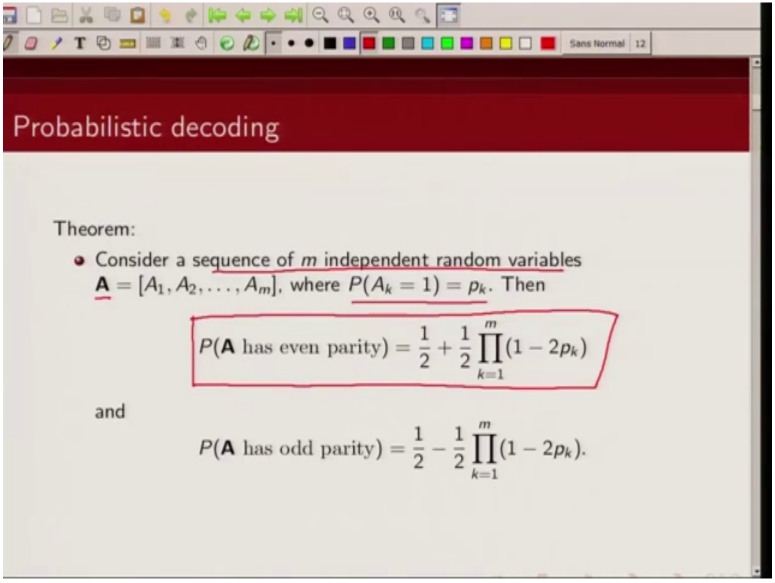
$$P(\mathbf{A} \text{ has even parity}) = \frac{1}{2} + \frac{1}{2}\prod_{k=1}^{m}(1-2p_k)$$

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 that if you have a sequence of m independent random variables which you denote by A, so A 1, A 2, A 3, A m are m independent random variable and probability of A A k being 1 is given p, Then probability that A has even parity is given by this expression

(Refer Slide Time 01:17)



Probabilistic decoding

Theorem:

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

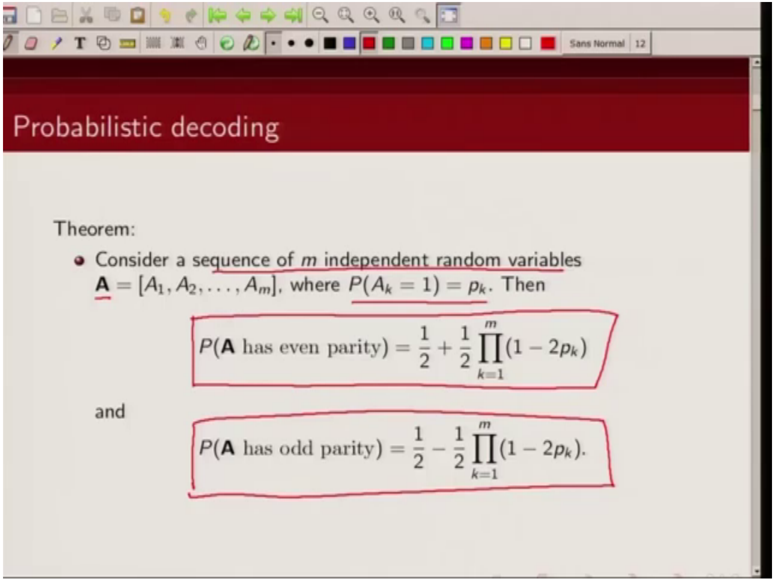
$$\boxed{P(\mathbf{A} \text{ has even parity}) = \frac{1}{2} + \frac{1}{2}\prod_{k=1}^{m}(1 - 2p_k)}$$

and

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

and probability that A has odd parity is given by this expression.

(Refer Slide Time 01:24)



Probabilistic decoding

Theorem:

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

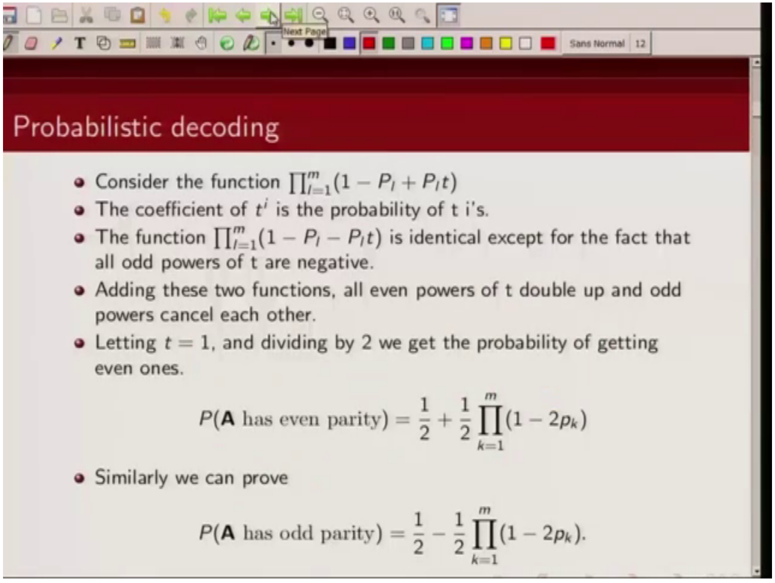
$$\boxed{P(\mathbf{A} \text{ has even parity}) = \frac{1}{2} + \frac{1}{2}\prod_{k=1}^{m}(1 - 2p_k)}$$

and

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

Now we will use, we will derive our expression for decoding algorithm based on these results. So let us first try to prove this result.

(Refer Slide Time 01:38)



Probabilistic decoding

- Consider the function $\prod_{l=1}^{m}(1 - P_l + P_l t)$
- The coefficient of t^i is the probability of t i's.
- The function $\prod_{l=1}^{m}(1 - P_l - P_l t)$ 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.

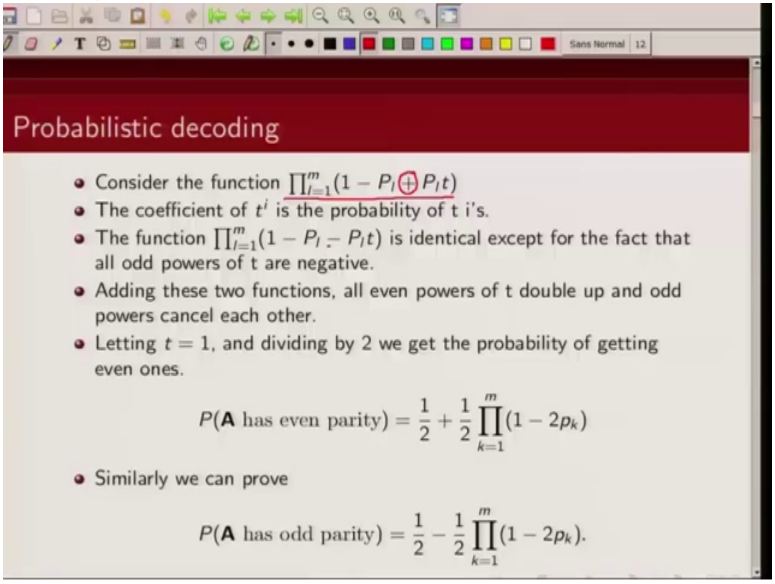
$$P(\mathbf{A} \text{ has even parity}) = \frac{1}{2} + \frac{1}{2}\prod_{k=1}^{m}(1 - 2p_k)$$

- Similarly we can prove

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

So let us consider the function of the form this, 1 minus p l plus p l t. So if we look at the coefficient of t here this will give us the probability of t is. Now we can similarly consider this function where this is, this plus has been replaced by minus

(Refer Slide Time 02:02)



Probabilistic decoding

- Consider the function $\underline{\prod_{l=1}^{m}(1 - P_l \oplus P_l t)}$
- The coefficient of t^i is the probability of t i's.
- The function $\prod_{l=1}^{m}(1 - P_l - P_l t)$ 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.

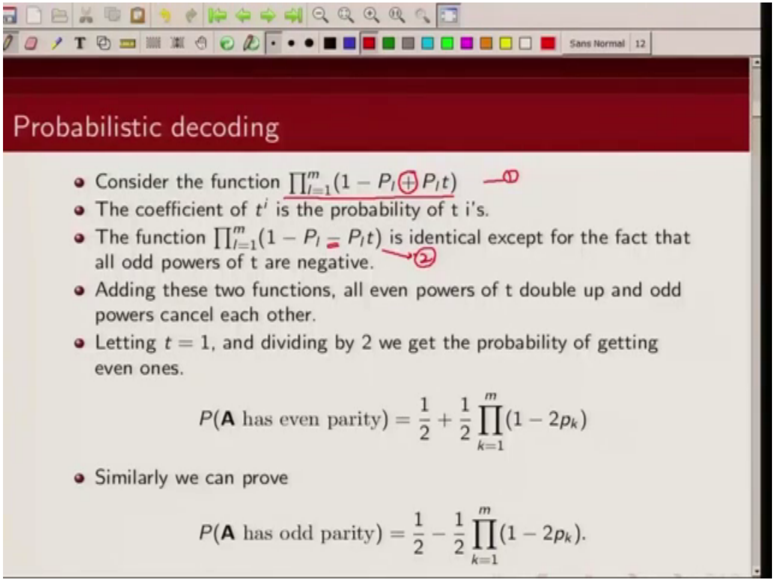
$$P(\mathbf{A} \text{ has even parity}) = \frac{1}{2} + \frac{1}{2}\prod_{k=1}^{m}(1 - 2p_k)$$

- Similarly we can prove

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

here. So this function is identical to this except that the odd powers of t will be, have minus in expansion. So if we expand this and we expand this, if we add

(Refer Slide Time 02:20)



Probabilistic decoding

- Consider the function $\underline{\prod_{l=1}^{m}(1-P_l \boxed{+} P_l t)}$ —①
- The coefficient of t^i is the probability of t i's.
- The function $\prod_{l=1}^{m}(1-P_l - P_l t)$ 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

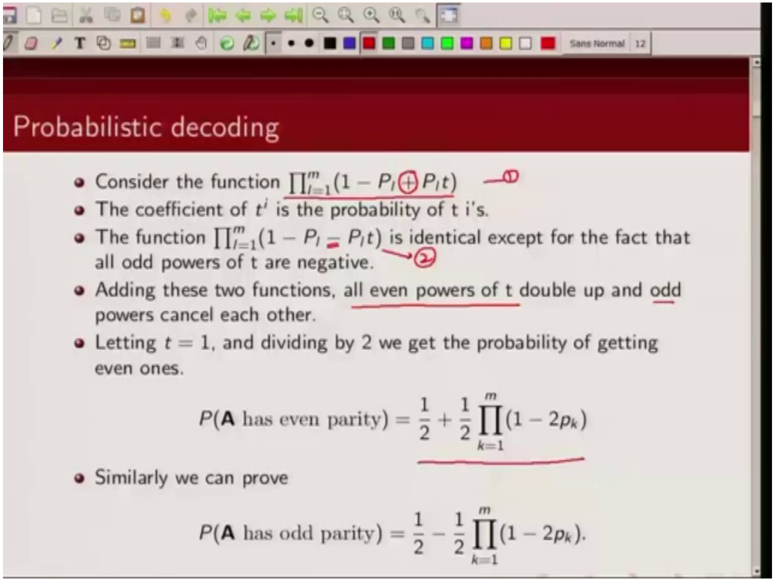
$$P(\mathbf{A}\text{ has odd parity}) = \frac{1}{2} - \frac{1}{2}\prod_{k=1}^{m}(1-2p_k).$$

both of them together what we will get is all the odd powers of t will go away. Ok so what we will be left will be even powers of t which are doubled up and odd powers have canceled out. If we put t equal to 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

(Refer Slide Time 02:52)

even. And similarly we can also prove

(Refer Slide Time 02:59)



Probabilistic decoding

- Consider the function $\underline{\prod_{l=1}^{m}(1 - P_l \oplus P_l t)}$ — ①
- The coefficient of t^i is the probability of t i's.
- The function $\prod_{l=1}^{m}(1 - P_l - P_l t)$ 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.

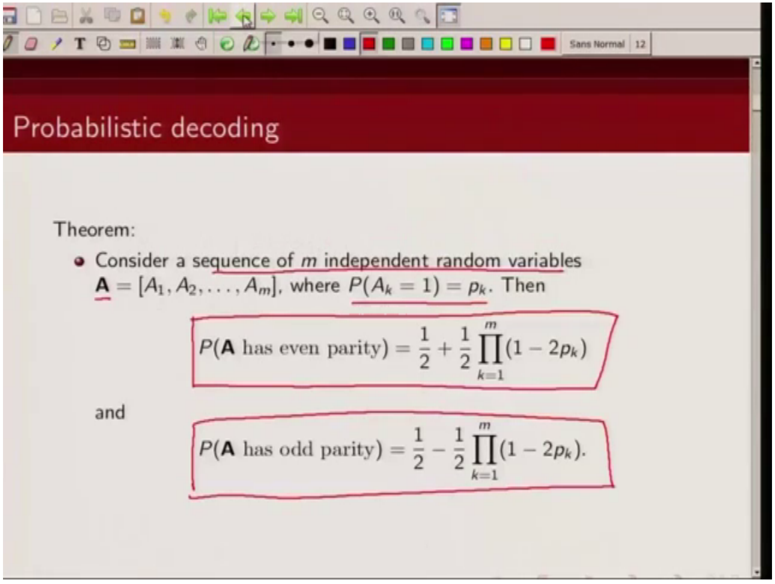
$$P(\mathbf{A} \text{ has even parity}) = \underline{\frac{1}{2} + \frac{1}{2}\prod_{k=1}^{m}(1 - 2p_k)}$$

- 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 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. Now the same result can also be proved

(Refer Slide Time 03:21)



Probabilistic decoding

Theorem:

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

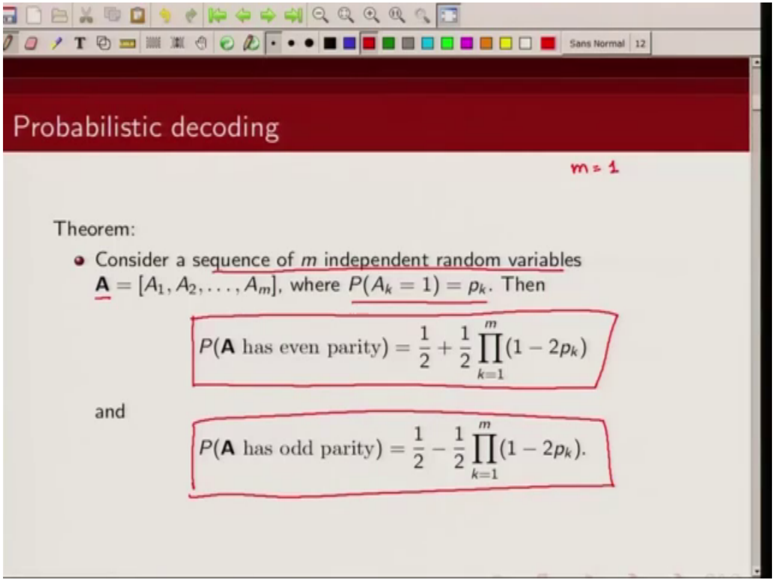
$$\boxed{P(\mathbf{A} \text{ has even parity}) = \frac{1}{2} + \frac{1}{2}\prod_{k=1}^{m}(1 - 2p_k)}$$

and

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

using mathematical induction. So if you want to prove it using mathematical induction, let's just say we take m equal to 1. If we take

(Refer Slide Time 03:32)



Probabilistic decoding

$m = 1$

Theorem:

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

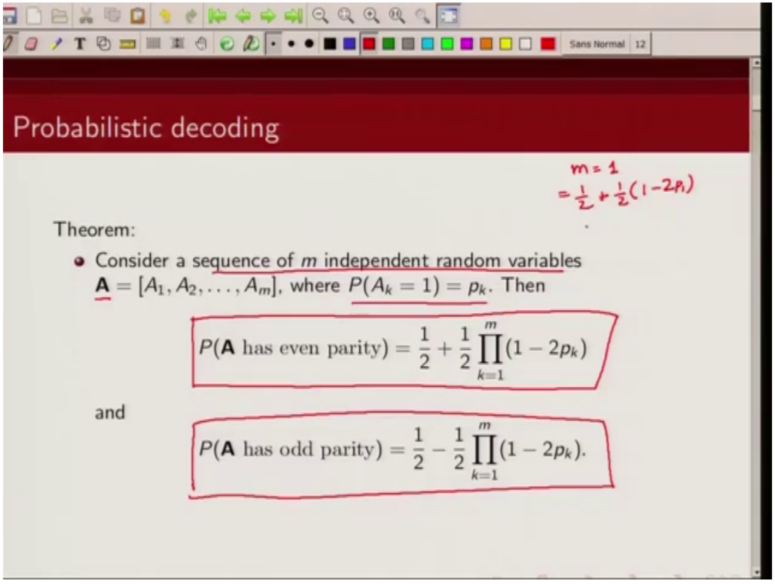
$$\boxed{P(\mathbf{A} \text{ has even parity}) = \frac{1}{2} + \frac{1}{2}\prod_{k=1}^{m}(1 - 2p_k)}$$

and

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

m equal to 1, what do we get here? For m equal to 1, we get basically half plus half one minus 2 p 1.

(Refer Slide Time 03:47)



Probabilistic decoding

$m = 1$

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

Theorem:

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

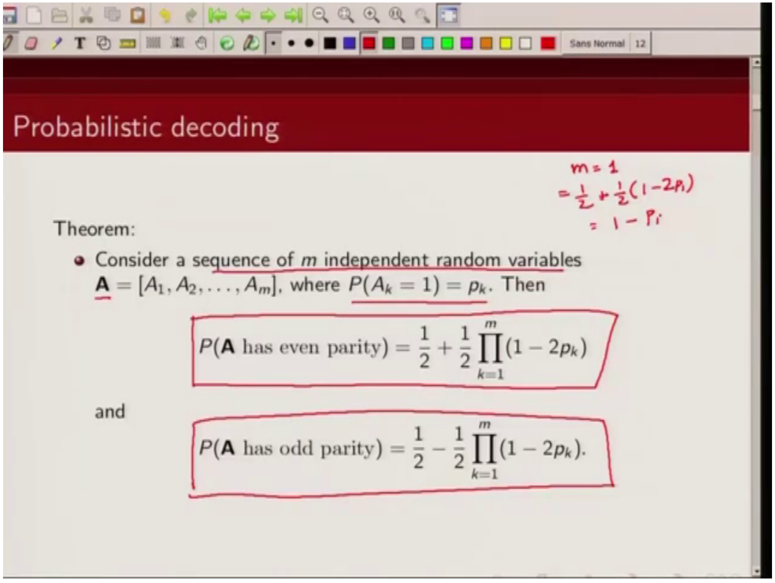
$$\boxed{P(\mathbf{A} \text{ has even parity}) = \frac{1}{2} + \frac{1}{2}\prod_{k=1}^{m}(1 - 2p_k)}$$

and

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

So this is nothing but 1 minus p 1.

(Refer Slide Time 03:55)



Probabilistic decoding

$m = 1$

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

$= 1 - p_1$

Theorem:

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

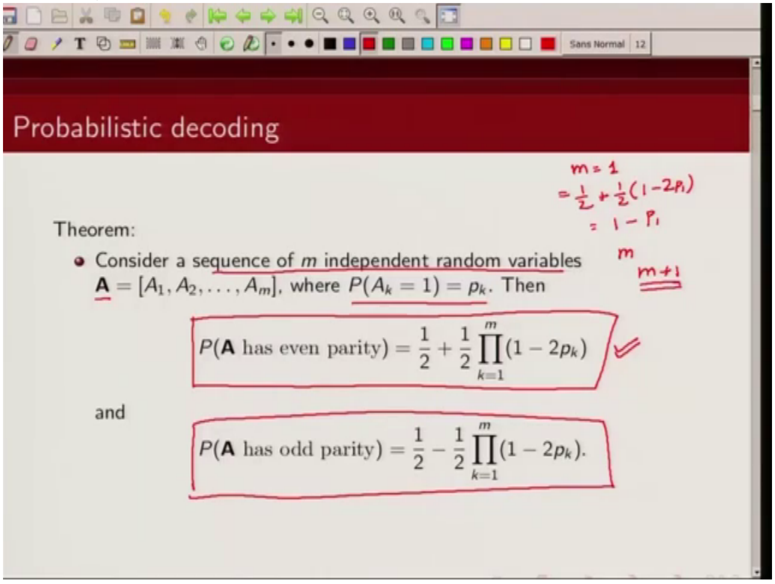
$$\boxed{P(\mathbf{A} \text{ has even parity}) = \frac{1}{2} + \frac{1}{2}\prod_{k=1}^{m}(1-2p_k)}$$

and

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

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 minus p 1, so this relation holds for m equal to 1. Now let us see it holds for also some m. Then we have to show that it also holds for m plus 1. Now if it holds for

(Refer Slide Time 04:19)



Probabilistic decoding

$m = 1$

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

$= 1 - p_1$

m

$\underline{\underline{m+1}}$

Theorem:

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

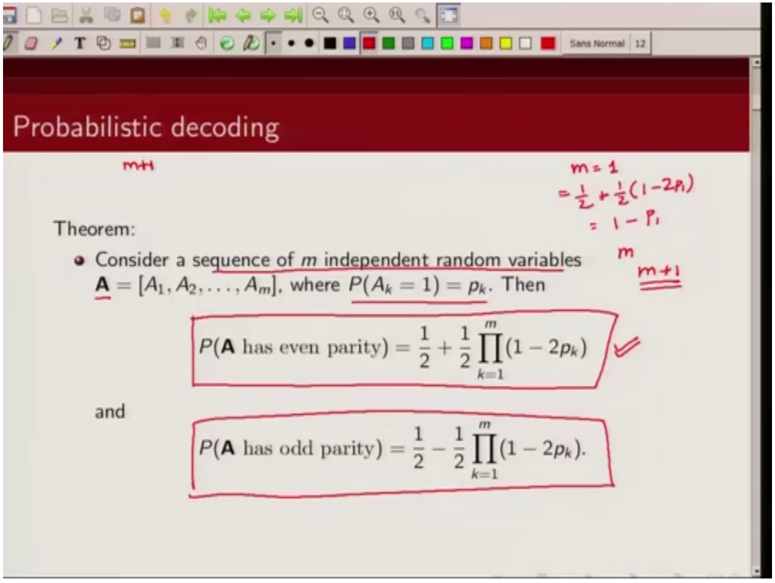
$$\boxed{P(\mathbf{A} \text{ has even parity}) = \frac{1}{2} + \frac{1}{2}\prod_{k=1}^{m}(1-2p_k)} \checkmark$$

and

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

m, to show that it holds for m plus 1, we have to find what's the probability that sequence of m plus 1, independent random variables have even parity. Now when they, m plus 1, independent random

(Refer Slide Time 04:35)



Probabilistic decoding

m+1

$m = 1$

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

$= 1 - p_1$

m

$\underline{\underline{m+1}}$

Theorem:

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

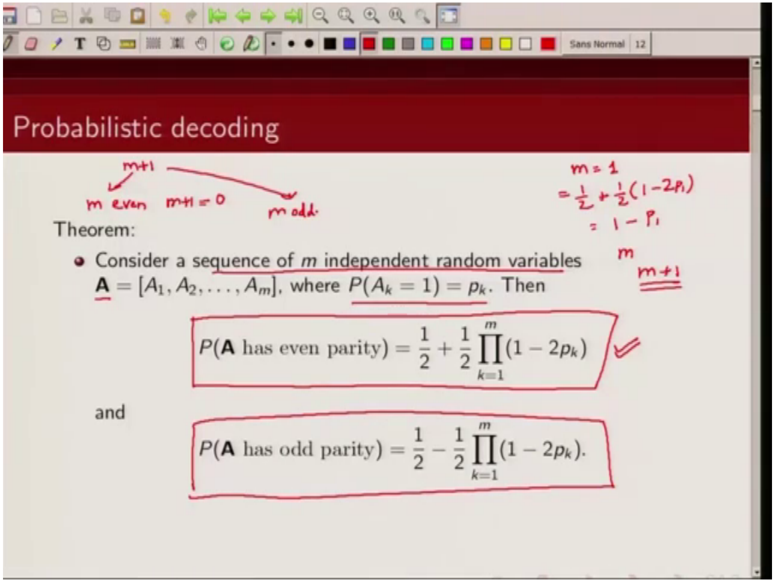
$$\boxed{P(\mathbf{A} \text{ has even parity}) = \frac{1}{2} + \frac{1}{2}\prod_{k=1}^{m}(1-2p_k)} \checkmark$$

and

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

variables have even parity? They will have even parity when m of them have even parity and mth bit that we get, mth bit is zero or sum of m independent variables, they give odd parity

(Refer Slide Time 05:03)



Probabilistic decoding

m+1

m even m+1 = 0

m odd.

$m = 1$

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

$= 1 - p_1$

m

$\underline{\underline{m+1}}$

Theorem:

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

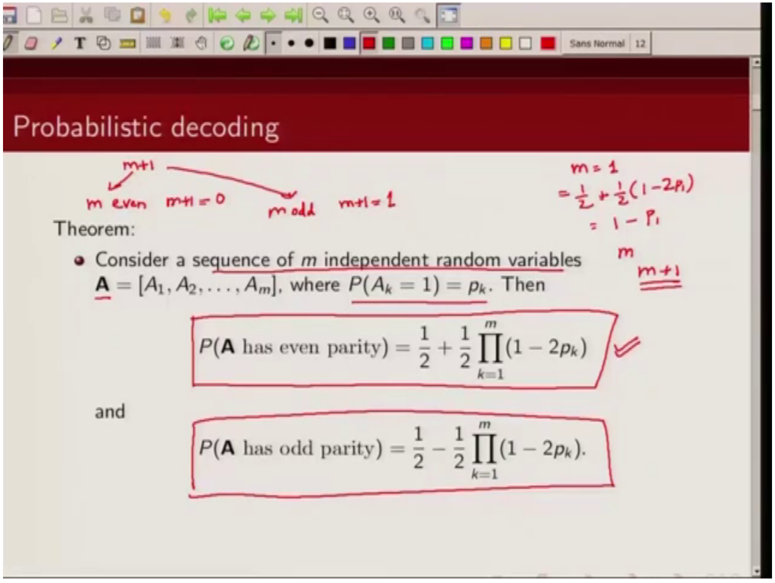
$$\boxed{P(\mathbf{A} \text{ has even parity}) = \frac{1}{2} + \frac{1}{2}\prod_{k=1}^{m}(1-2p_k)} \checkmark$$

and

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

the mth bit that we receive is actually 1. So then

(Refer Slide Time 05:08)



Probabilistic decoding

m+1

m even m+1 = 0

m odd m+1 = 1

m = 1

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

$= 1 - p_1$

m

m+1

Theorem:

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

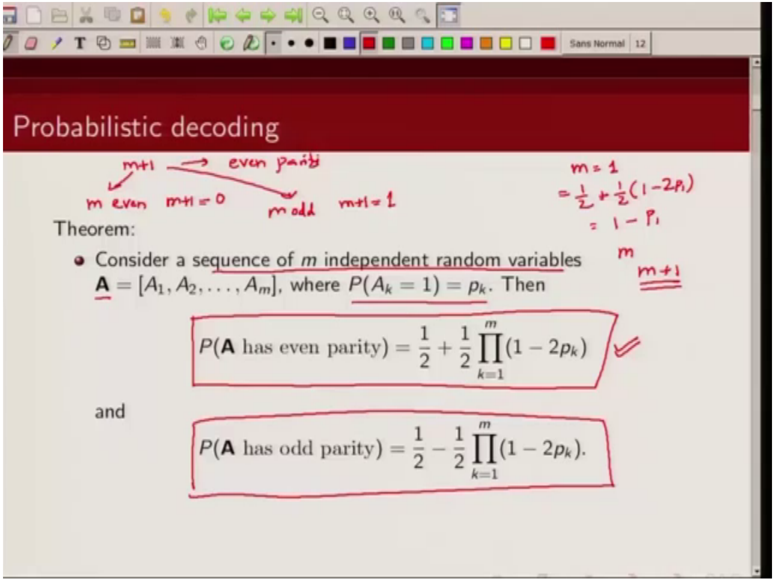
$$\boxed{P(\mathbf{A} \text{ has even parity}) = \frac{1}{2} + \frac{1}{2}\prod_{k=1}^{m}(1-2p_k)} \checkmark$$

and

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

we will get m plus 1 as even parity.

(Refer Slide Time 05:17)



Probabilistic decoding

m+1 → even parity

m even m+1 = 0

m odd m+1 = 1

m = 1

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

$= 1 - p_1$

m

m+1

Theorem:

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

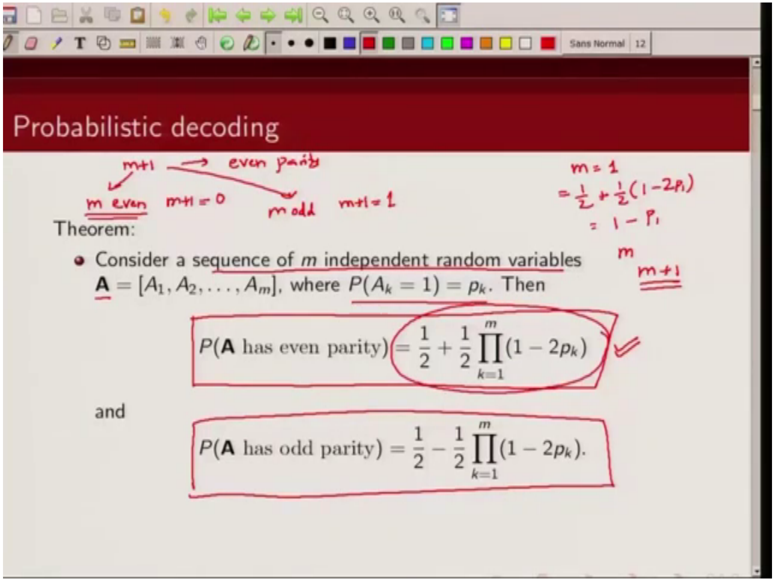
$$\boxed{P(\mathbf{A} \text{ has even parity}) = \frac{1}{2} + \frac{1}{2}\prod_{k=1}^{m}(1-2p_k)} \checkmark$$

and

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

Is it clear? So there are 2 cases, two ways in which we can get even parity when we are considering m plus 1 independent random variables. Either m of them had even parity and m plus 1 is actually zero or m of them add up to odd parity and m plus 1 is also odd parity, Ok. So what is this probability of m being even? That's given by this. What's the

(Refer Slide Time 05:52)



Probabilistic decoding

$m+1 \rightarrow$ even parity

m even $\quad m+1 = 0$ $\qquad$ m odd $\quad m+1 = 1$

$m = 1$

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

$= 1 - p_1$

m

$m+1$

Theorem:

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

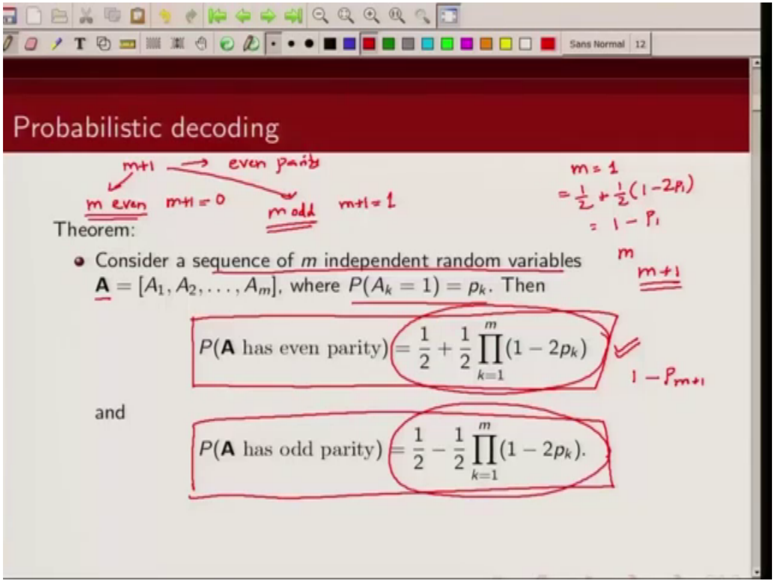
$$P(\mathbf{A}\text{ has even parity}) = \frac{1}{2} + \frac{1}{2}\prod_{k=1}^{m}(1-2p_k)$$

and

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

probability that m plus 1 is zero? That is given by 1 minus p m plus 1. What is the probability that m of them is odd? That's given by this probability.

(Refer Slide Time 06:11)



Probabilistic decoding

$m+1 \rightarrow$ even parity

m even $\quad m+1 = 0$ $\qquad$ m odd $\quad m+1 = 1$

$m = 1$

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

$= 1 - p_1$

m

$m+1$

Theorem:

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

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

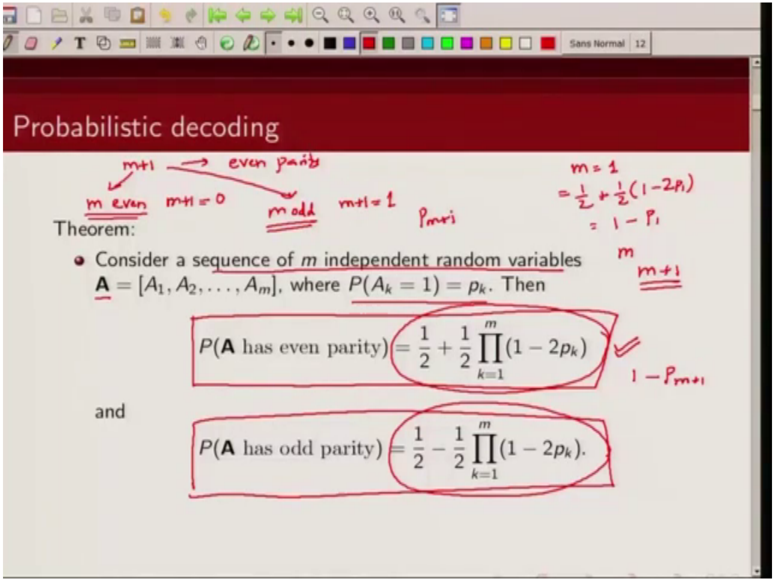
$1 - p_{m+1}$

and

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

And what's the probability that m plus 1 random variable is 1? That probability is given by p times m plus 1. So the overall probability

(Refer Slide Time 06:22)



Probabilistic decoding

m+1 → even parity

m even m+1 = 0

m odd m+1 = 1

P_{m+1}

$m = 1$

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

$= 1 - p_1$

m

m+1

Theorem:

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

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

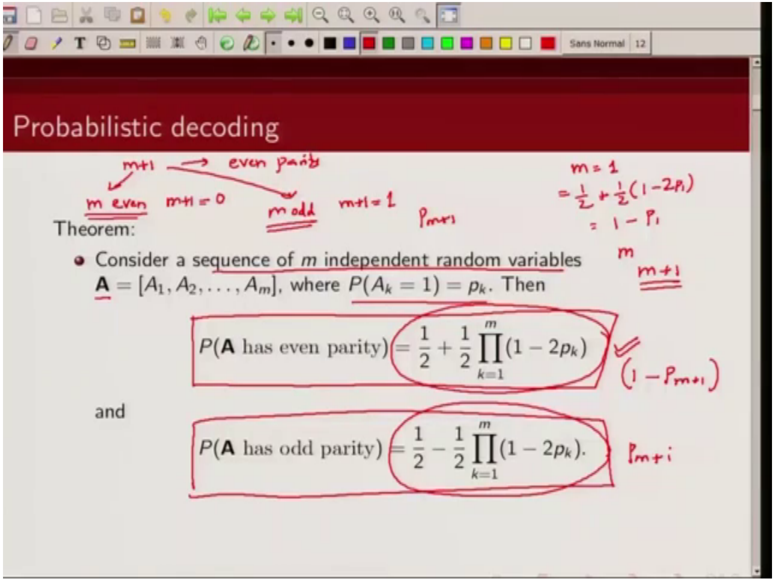
$1 - P_{m+1}$

and

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

will be this multiplied by this plus this multiplied by p m plus 1. And if you simplify this

(Refer Slide Time 06:33)



Probabilistic decoding

m+1 → even parity

m even m+1 = 0

m odd m+1 = 1

P_{m+1}

$m = 1$

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

$= 1 - p_1$

m

m+1

Theorem:

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

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

$(1 - P_{m+1})$

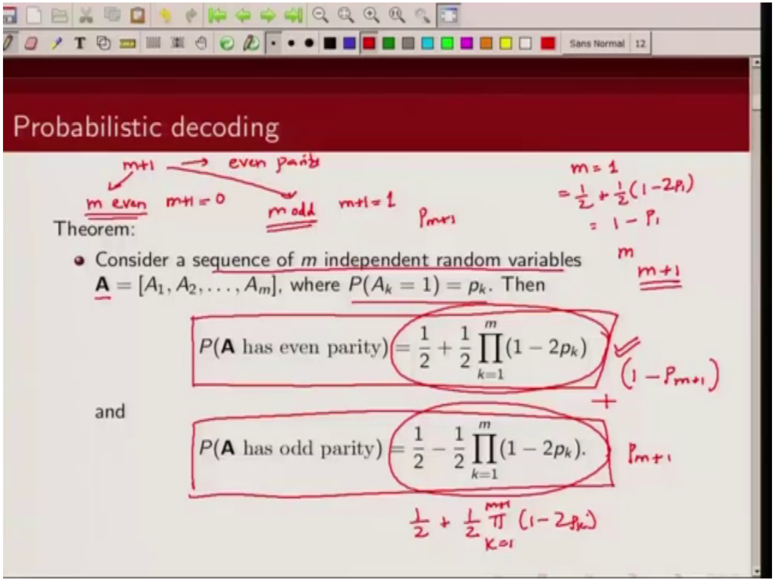
and

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

P_{m+1}

we can show that this plus this will come out to be 1 plus 1 k equal to 1 m plus 1 1 minus 2 p k. So details of the calculation I am just leaving it

(Refer Slide Time 06:51)



Probabilistic decoding

m+1 → even parity

m even m+1 = 0 m odd m+1 = 1 p_{m+1}

m = 1

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

$= 1 - p_1$

m

m+1

Theorem:

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

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

✓ $(1 - p_{m+1})$

+

and

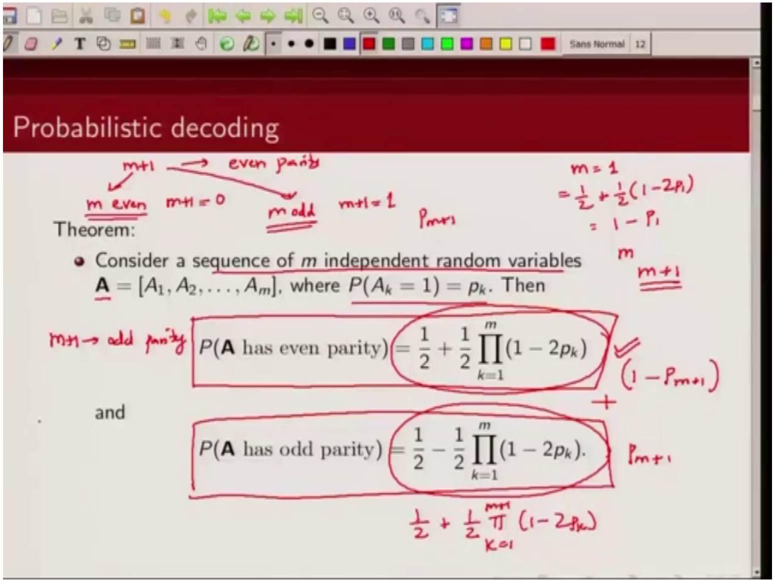
$$P(\mathbf{A} \text{ has odd parity}) = \frac{1}{2} - \frac{1}{2}\prod_{k=1}^{m}(1-2p_k).$$

p_{m+1}

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

but this is how, using mathematical induction we are going to prove that this also holds true for m plus 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 equal to 1, this is nothing but p 1, so that it holds true for m equal to 1. Assume it holds for m, then we have to show it also holds for m plus 1. Now for m plus 1 to have odd parity, two ways,

(Refer Slide Time 07:31)



Probabilistic decoding

m+1 → even parity

m even m+1 = 0 m odd m+1 = 1 p_{m+1}

m = 1

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

$= 1 - p_1$

m

m+1

Theorem:

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

m+1 → odd parity

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

✓ $(1 - p_{m+1})$

+

and

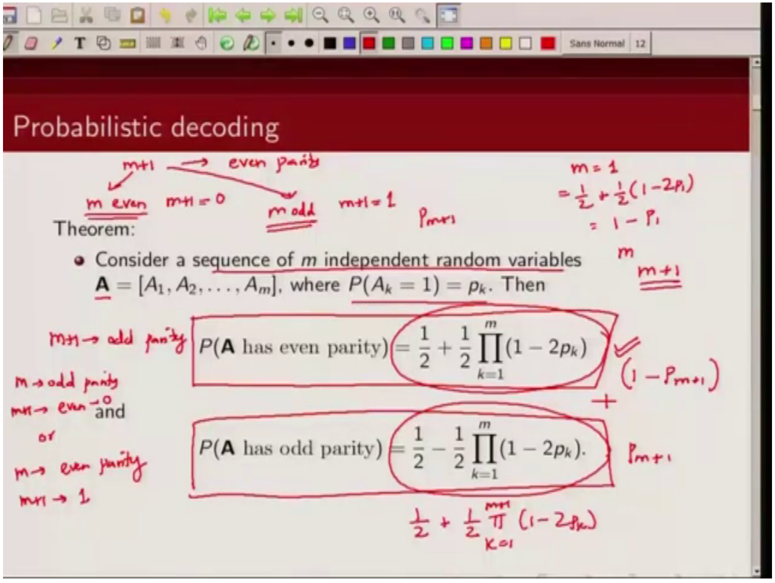
$$P(\mathbf{A} \text{ has odd parity}) = \frac{1}{2} - \frac{1}{2}\prod_{k=1}^{m}(1-2p_k).$$

p_{m+1}

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

m has odd parity and the m plus 1 bit that we get is even parity is zero basically, is zero. Or m has even parity and m plus 1 bit that we get is actually 1.

(Refer Slide Time 07:52)



Probabilistic decoding

Theorem:

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

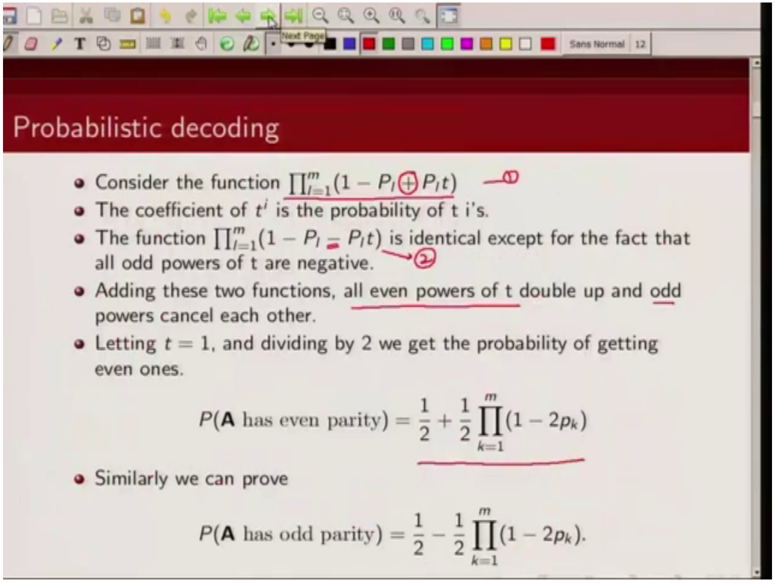
$$P(\mathbf{A} \text{ has even parity}) = \frac{1}{2} + \frac{1}{2}\prod_{k=1}^{m}(1 - 2p_k)$$

and

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

So similar to this case we can also find out these probability and we add them up we can show that this will be equal to half minus half product from k equal to 1 to m plus 1 of this. So we can say that probability that a has even parity is given by this and probability that a has, so sum of these m random variables have odd parity is given by this. So these are the crucial expressions that we would be using in our decoding algorithm.

(Refer Slide Time 08:30)



Probabilistic decoding

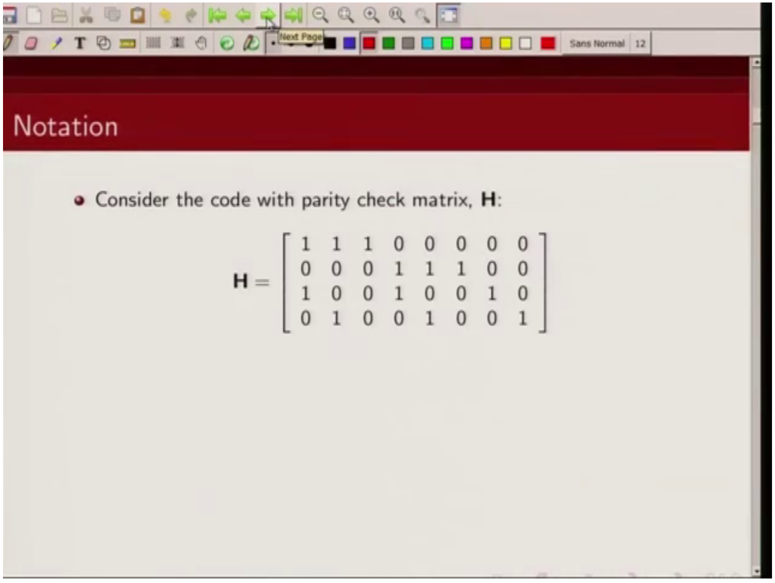
- Consider the function $\prod_{l=1}^{m}(1 - P_l + P_l t)$
- The coefficient of t^i is the probability of t i's.
- The function $\prod_{l=1}^{m}(1 - P_l - P_l t)$ is identical except for the fact that all odd powers of t are negative.
- Adding these two functions, all even powers of t double up and odd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

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

- 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 08:33)



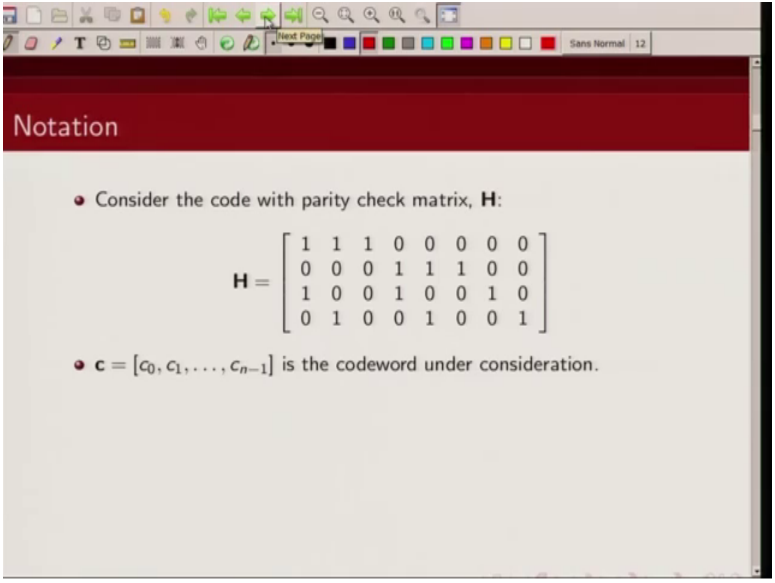
Notation

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

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

So let us consider an example. So this is our parity check matrix. We will first

(Refer Slide Time 08:39)



Notation

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

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

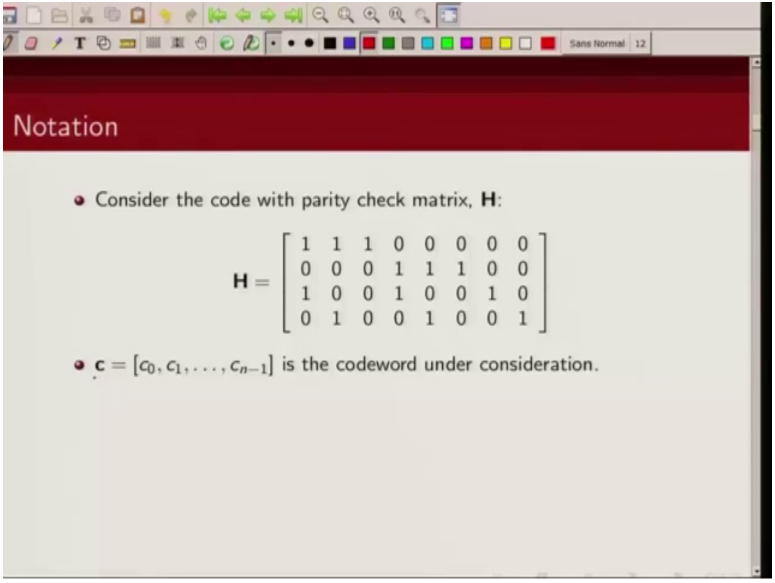
- $\mathbf{c} = [c_0, c_1, \ldots, c_{n-1}]$ is the codeword under consideration.

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

(Refer Slide Time 08:52)

example. So we are

(Refer Slide Time 08:54)



Notation

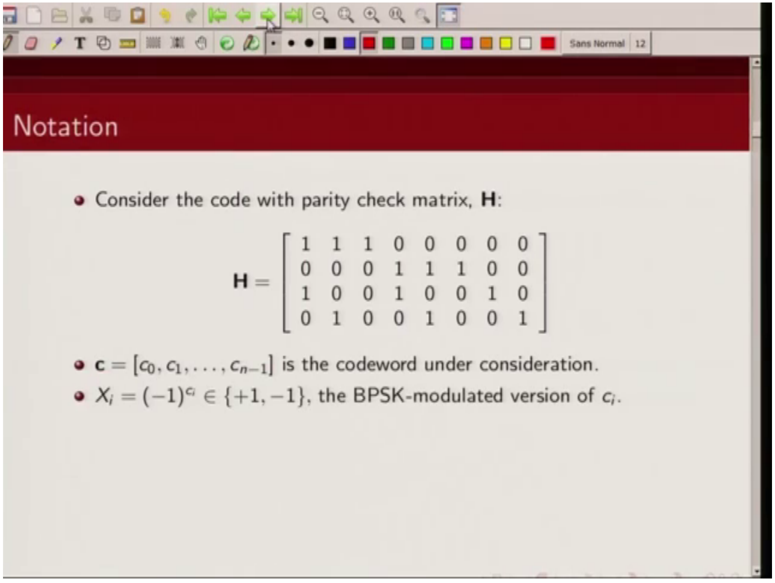
- Consider the code with parity check matrix, **H**:

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

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

using c to denote our codeword. So this is an n bit codeword, c 0, c 1, c 2, c n. That's our codeword.

(Refer Slide Time 09:04)



Notation

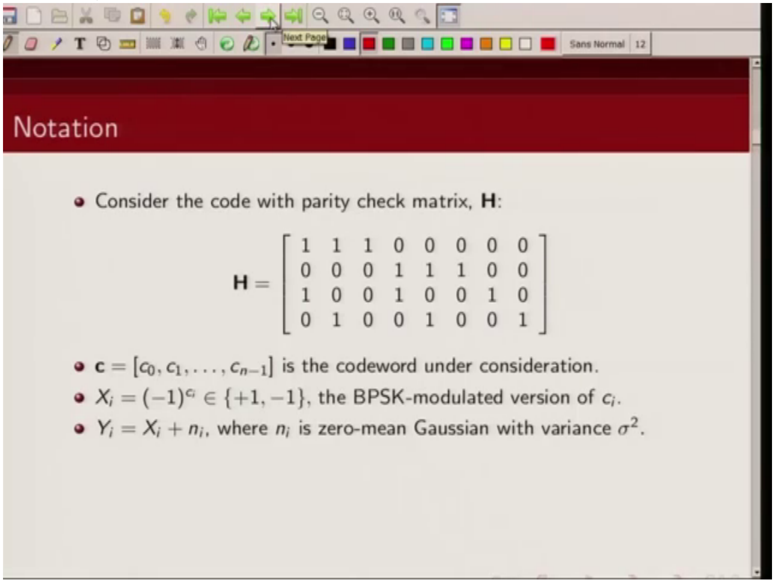
- Consider the code with parity check matrix, $\mathbf{H}$:

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

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

Now this codeword is modulated using b p s k modulations so we are mapping 0 to 1, we are mapping to minus 1. So that's your b p s k modulated version of the code bits denoted by x of i.

(Refer Slide Time 09:25)



Notation

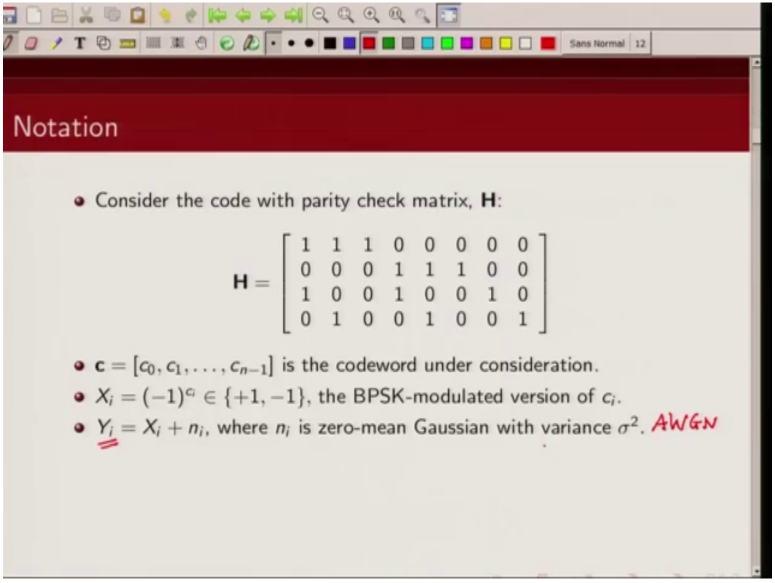
- Consider the code with parity check matrix, $\mathbf{H}$:

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

- $\mathbf{c} = [c_0, c_1, \ldots, c_{n-1}]$ 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 σ^2.

y of i is your received modulated codeword. So this is, we are considering an additive white gaussian noise channel with

(Refer Slide Time 09:38)



Notation

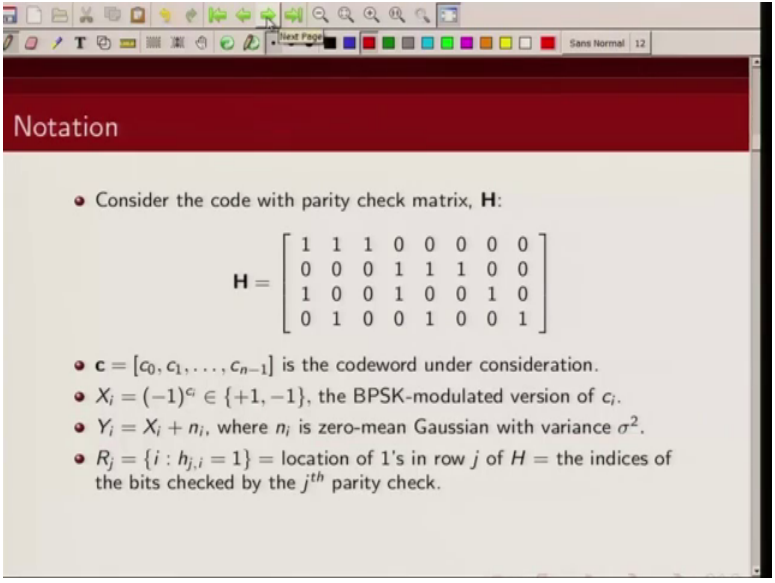
- Consider the code with parity check matrix, $\mathbf{H}$:

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

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

noise variance given by n square and this is zero mean gaussian noise.

(Refer Slide Time 09:46)



Notation

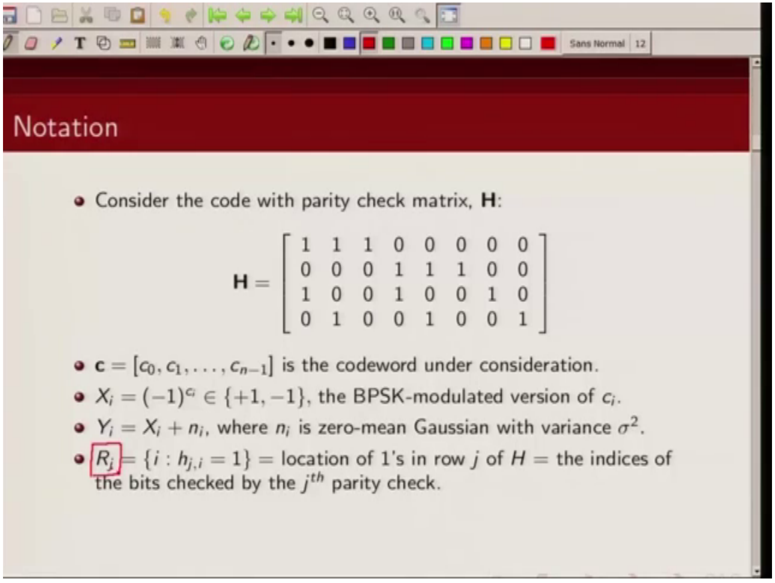
- Consider the code with parity check matrix, $\mathbf{H}$:

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

- $\mathbf{c} = [c_0, c_1, \ldots, c_{n-1}]$ 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 σ^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^{th} parity check.

We use this notation R j

(Refer Slide Time 09:51)



Notation

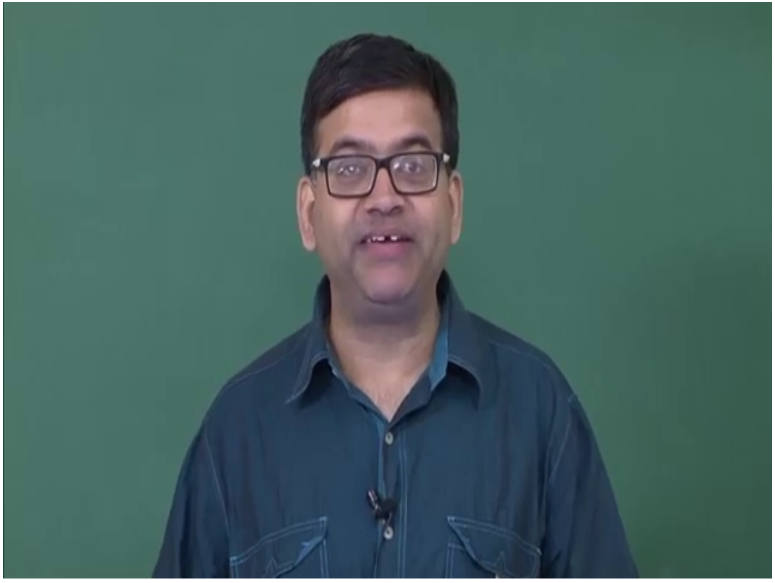
- Consider the code with parity check matrix, **H**:

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

- $\mathbf{c} = [c_0, c_1, \ldots, c_{n-1}]$ 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 σ^2.
- $\boxed{R_j} = \{i : h_{j,i} = 1\}$ = location of 1's in row j of H = the indices of the bits checked by the j^{th} parity check.

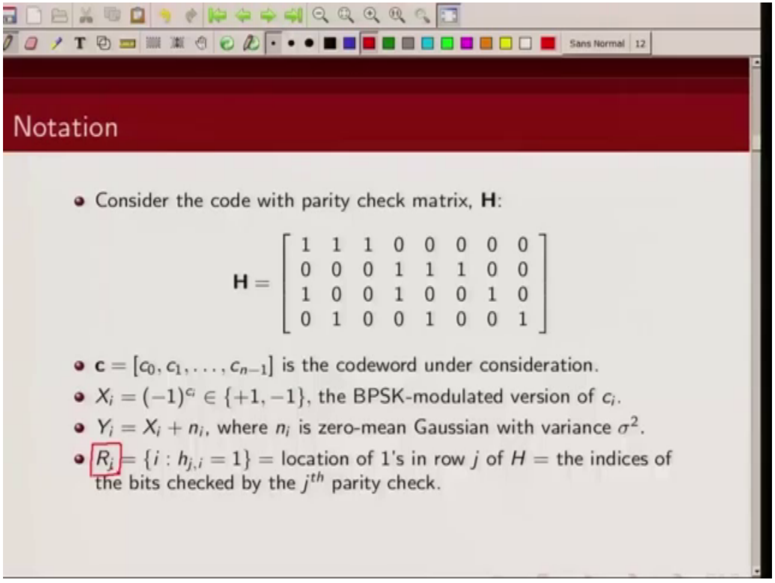
to denote the location of 1s in row j. Now what location of 1s in the parity check

(Refer Slide Time 10:05)



matrix denote? It denotes the bits that are participating in the parity check equation. So let's

(Refer Slide Time 10:14)



Notation

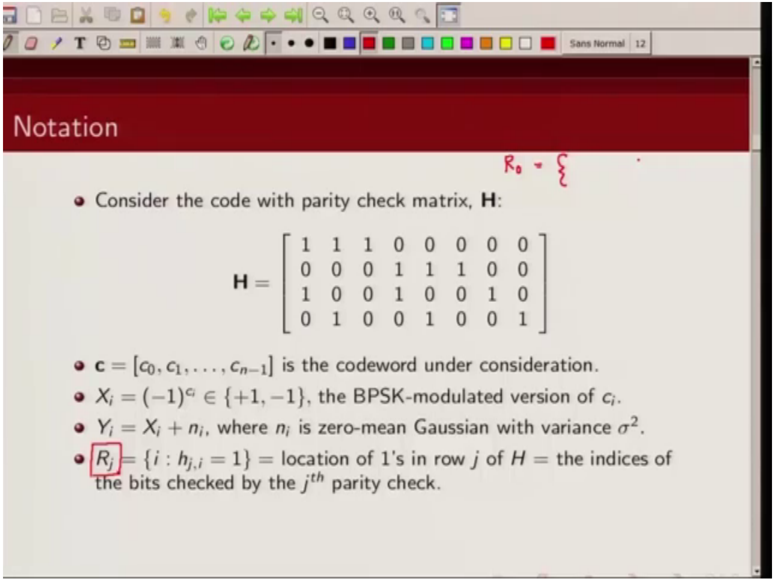
- Consider the code with parity check matrix, **H**:

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

- $\mathbf{c} = [c_0, c_1, \ldots, c_{n-1}]$ 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 σ^2.
- $\boxed{R_j} = \{i : h_{j,i} = 1\}$ = location of 1's in row j of H = the indices of the bits checked by the j^{th} parity check.

take this example. What is R 0? The R 0, so the R 0

(Refer Slide Time 10:21)



$R_0 = \{$

Notation

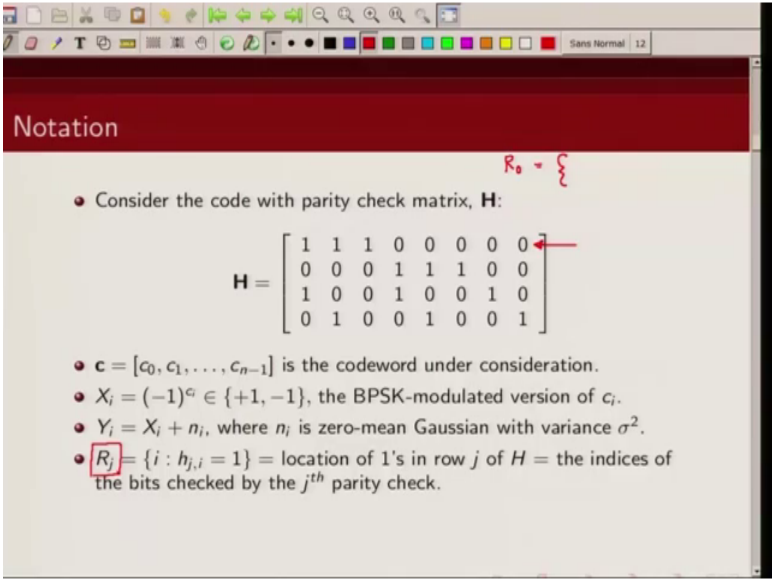
- Consider the code with parity check matrix, **H**:

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

- $\mathbf{c} = [c_0, c_1, \ldots, c_{n-1}]$ 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 σ^2.
- $\boxed{R_j} = \{i : h_{j,i} = 1\}$ = location of 1's in row j of H = the indices of the bits checked by the j^{th} parity check.

corresponds to the 0th row

(Refer Slide Time 10:24)



Notation

$R_0 = \{$

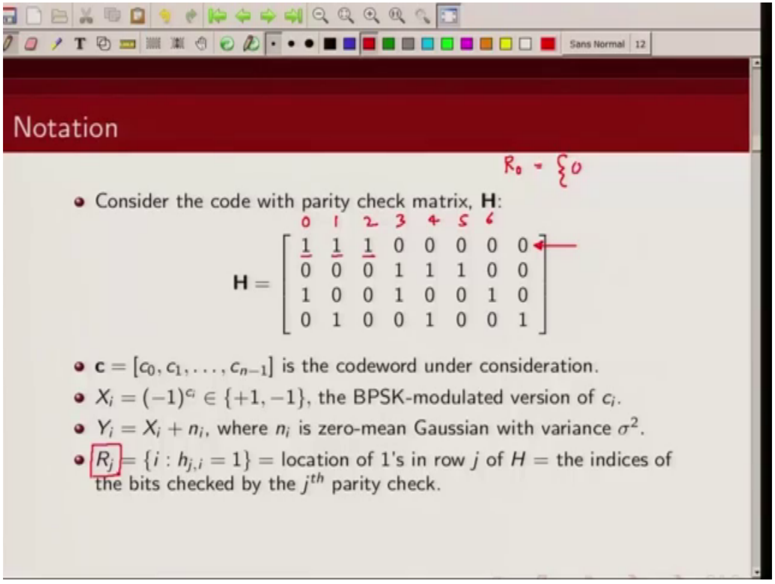
- Consider the code with parity check matrix, $\mathbf{H}$:

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

- $\mathbf{c} = [c_0, c_1, \ldots, c_{n-1}]$ 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 σ^2.
- $\boxed{R_j} = \{i : h_{j,i} = 1\}$ = location of 1's in row j of H = the indices of the bits checked by the j^{th} parity check.

which is this row. Now look at the bits that are 1 here, 1, 2, 3 these are the bits which are 1. So R 0 corresponds to, let's just label them 0, 1, 2, 3, 4, 5, 6. So R 0 corresponds to

(Refer Slide Time 10:41)



Notation

$R_0 = \{0$

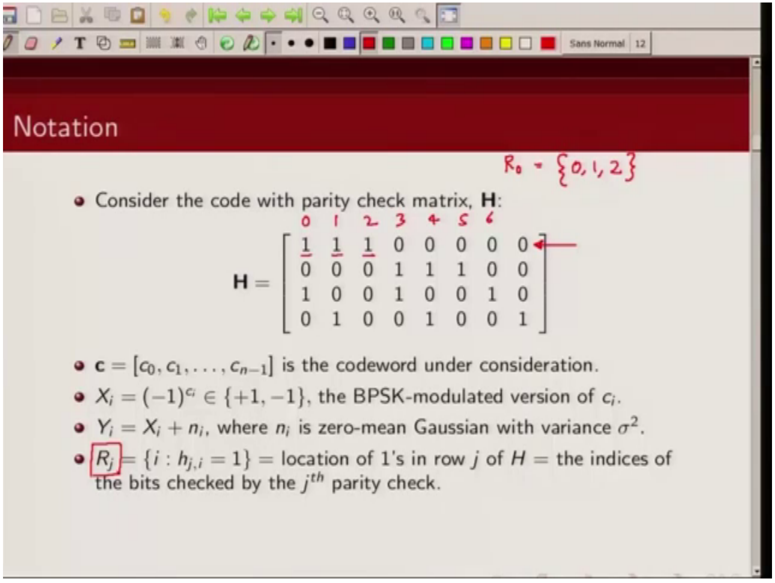
- Consider the code with parity check matrix, $\mathbf{H}$:

$$\mathbf{H} = \begin{array}{cccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & \\ \underline{1} & \underline{1} & \underline{1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{array}$$

- $\mathbf{c} = [c_0, c_1, \ldots, c_{n-1}]$ 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 σ^2.
- $\boxed{R_j} = \{i : h_{j,i} = 1\}$ = location of 1's in row j of H = the indices of the bits checked by the j^{th} parity check.

to 0, 1 and 2.

(Refer Slide Time 10:46)



Notation

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

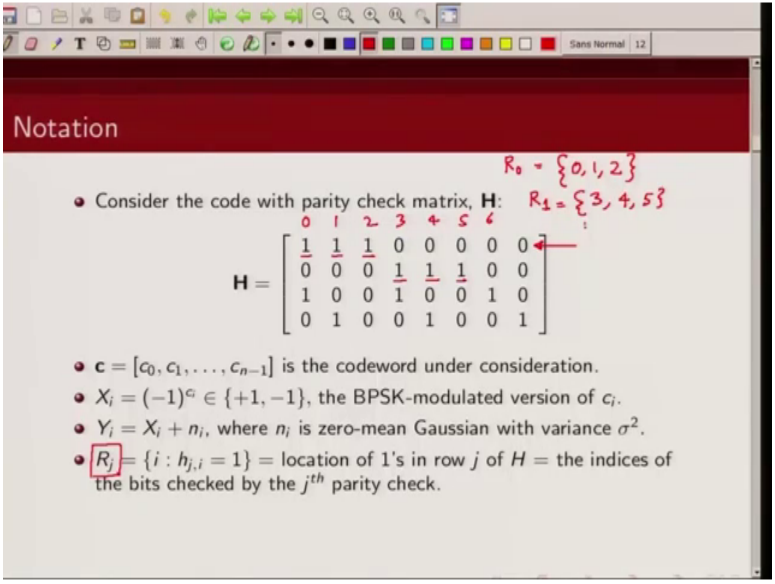
- Consider the code with parity check matrix, $\mathbf{H}$:

$$\mathbf{H} = \left[\begin{array}{cccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & \\ \underline{1} & \underline{1} & \underline{1} & 0 & 0 & 0 & 0 & 0 \leftarrow \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{array}\right]$$

- $\mathbf{c} = [c_0, c_1, \ldots, c_{n-1}]$ 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 σ^2.
- $\boxed{R_j} = \{i : h_{j,i} = 1\}$ = location of 1's in row j of H = the indices of the bits checked by the j^{th} parity check.

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. R 1 will be 3,4 and 5.

(Refer Slide Time 11:07)



Notation

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

$R_1 = \{3, 4, 5\}$

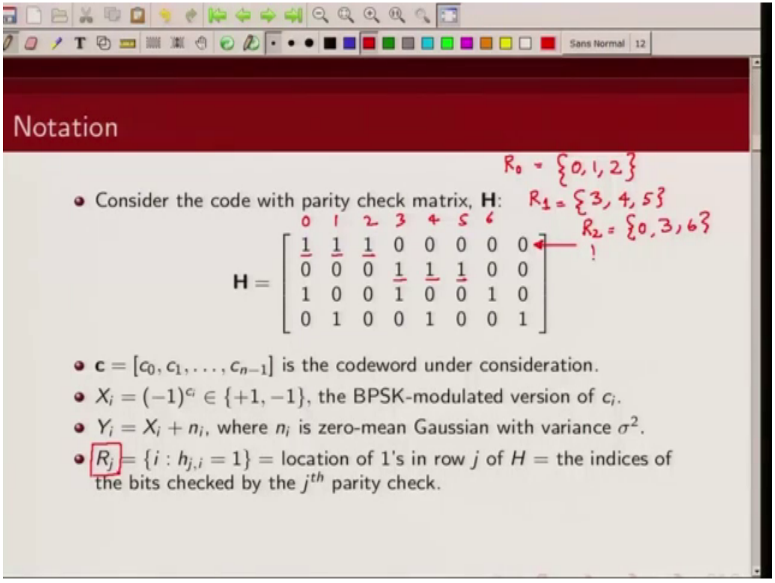
- Consider the code with parity check matrix, $\mathbf{H}$:

$$\mathbf{H} = \left[\begin{array}{cccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & \\ \underline{1} & \underline{1} & \underline{1} & 0 & 0 & 0 & 0 & 0 \leftarrow \\ 0 & 0 & 0 & \underline{1} & \underline{1} & \underline{1} & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{array}\right]$$

- $\mathbf{c} = [c_0, c_1, \ldots, c_{n-1}]$ 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 σ^2.
- $\boxed{R_j} = \{i : h_{j,i} = 1\}$ = location of 1's in row j of H = the indices of the bits checked by the j^{th} parity check.

R 2 will be 0, 3 and 6.And

(Refer Slide Time 11:18)



Notation

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

- Consider the code with parity check matrix, $\mathbf{H}$: $R_1 = \{3,4,5\}$

$R_2 = \{0,3,6\}$

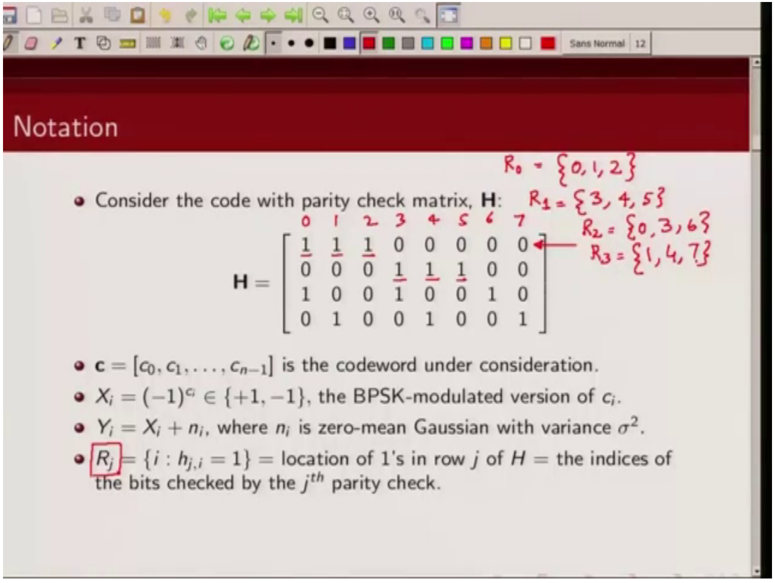
!

$$\mathbf{H} = \begin{array}{cccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & \\ \underline{1} & \underline{1} & \underline{1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \underline{1} & \underline{1} & \underline{1} & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{array}$$

- $\mathbf{c} = [c_0, c_1, \ldots, c_{n-1}]$ 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 σ^2.
- $\boxed{R_j} = \{i : h_{j,i} = 1\}$ = location of 1's in row j of H = the indices of the bits checked by the j^{th} parity check.

R 3 will be 1, 4, and 7

(Refer Slide Time 11:29)



Notation

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

- Consider the code with parity check matrix, $\mathbf{H}$: $R_1 = \{3,4,5\}$

$R_2 = \{0,3,6\}$

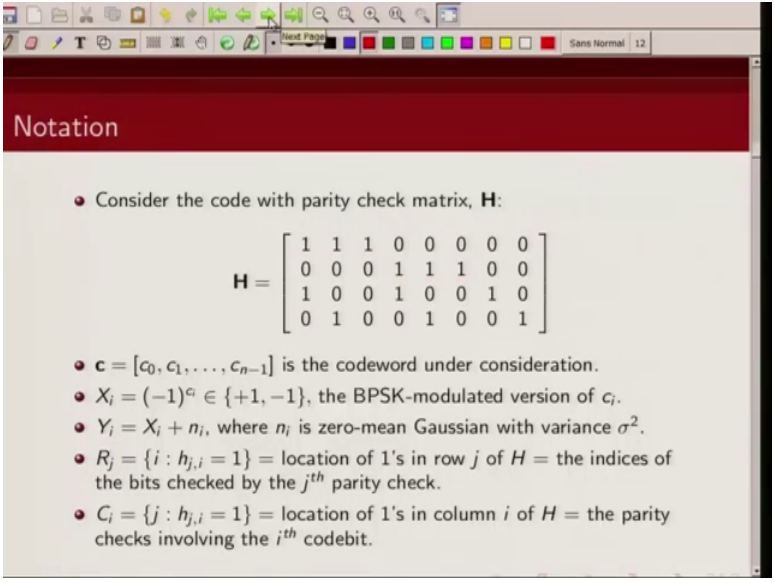
$R_3 = \{1,4,7\}$

$$\mathbf{H} = \begin{array}{cccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \underline{1} & \underline{1} & \underline{1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \underline{1} & \underline{1} & \underline{1} & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{array}$$

- $\mathbf{c} = [c_0, c_1, \ldots, c_{n-1}]$ 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 σ^2.
- $\boxed{R_j} = \{i : h_{j,i} = 1\}$ = location of 1's in row j of H = the indices of the bits checked by the j^{th} parity check.

So that's your R j.

(Refer Slide Time 11:34)



Notation

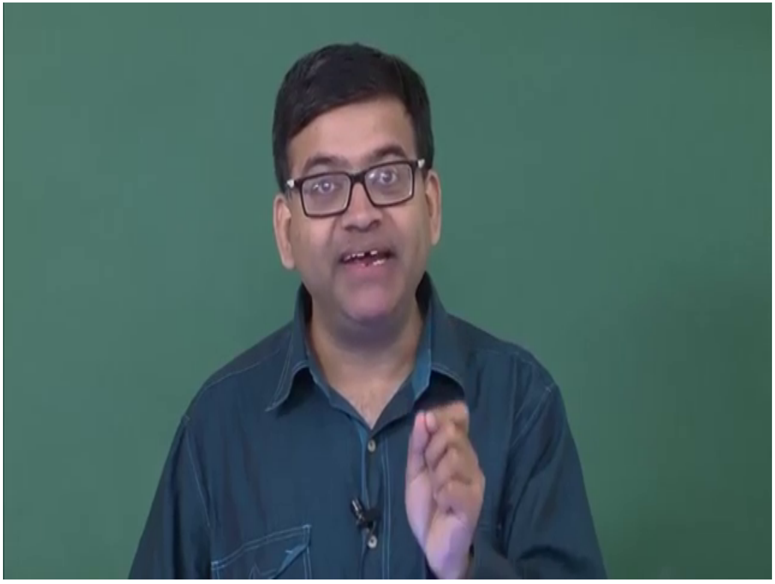
- Consider the code with parity check matrix, $\mathbf{H}$:

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

- $\mathbf{c} = [c_0, c_1, \ldots, c_{n-1}]$ 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 σ^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^{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^{th} codebit.

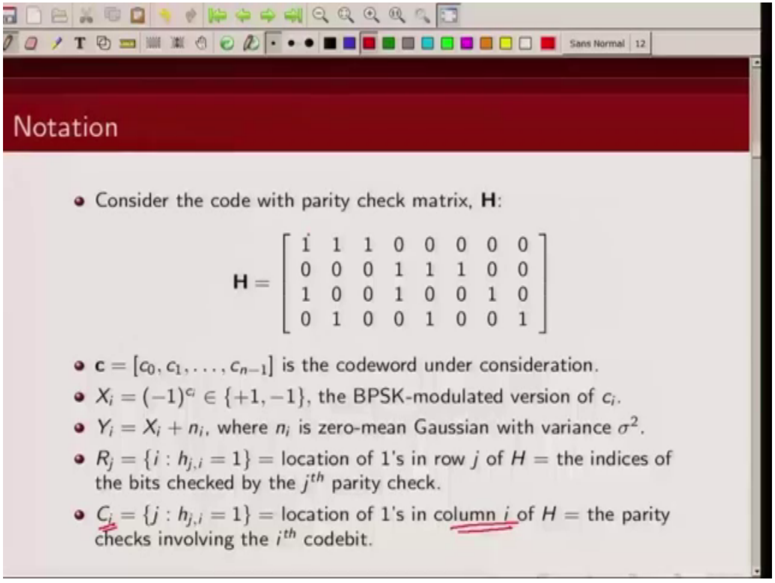
Now we use the notation c i to denote the locations of 1s in column i. Now what does location in column i denotes? It denotes that ith bit, that participating in how many

(Refer Slide Time 11:56)



parity check equations, which parity check equation. So let us

(Refer Slide Time 12:00)



Notation

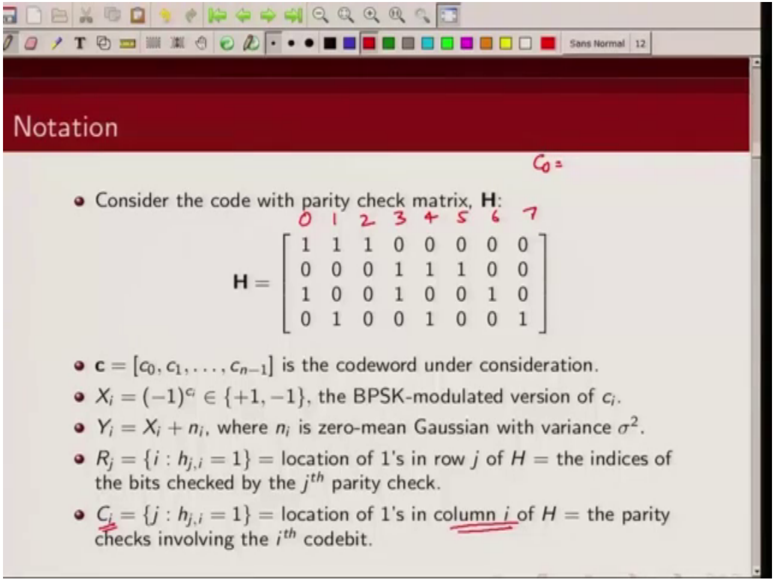
- Consider the code with parity check matrix, $\mathbf{H}$:

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

- $\mathbf{c} = [c_0, c_1, \ldots, c_{n-1}]$ 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 σ^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^{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^{th} codebit.

look at c 0.So this was again 0, 1, 2, 3, 4, 5, 6, 7. So c 0

(Refer Slide Time 12:10)



Notation

$C_0 =$

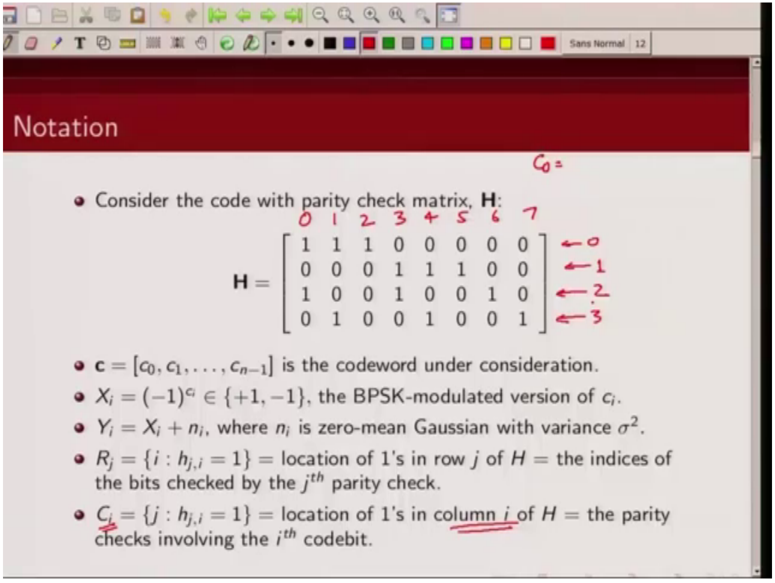
- Consider the code with parity check matrix, $\mathbf{H}$:

$$\mathbf{H} = \begin{array}{cccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \hline 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{array}$$

- $\mathbf{c} = [c_0, c_1, \ldots, c_{n-1}]$ 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 σ^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^{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^{th} codebit.

is what? c 0 is, again this is, let's call it 0,1,2, and 3. So this is 0th parity check equation,

(Refer Slide Time 12:22)



Notation

$C_0 =$

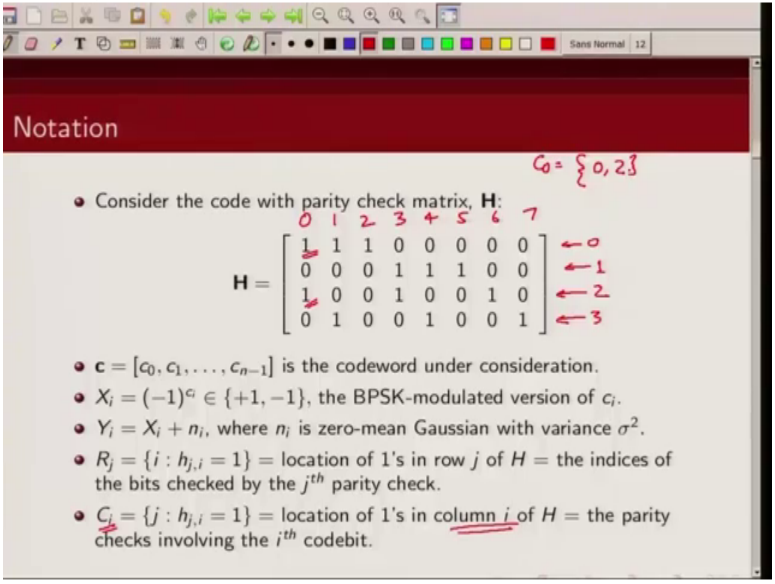
- Consider the code with parity check matrix, **H**:

$$\mathbf{H} = \begin{array}{ccccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & \leftarrow 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & \leftarrow 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & \leftarrow 2 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & \leftarrow 3 \end{array}$$

- $\mathbf{c} = [c_0, c_1, \ldots, c_{n-1}]$ 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 σ^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^{th} parity check.
- $C_i = \{j : h_{j,i} = 1\}$ = location of 1's in $\underline{\text{column } i}$ of H = the parity checks involving the i^{th} codebit.

1,2 and 3. So bit number, so if we look at column 0, so you can see this bit is participating in 0th parity check equation and second parity check equation. c 0 will be 0 and 2.

(Refer Slide Time 12:40)



Notation

$C_0 = \{0, 2\}$

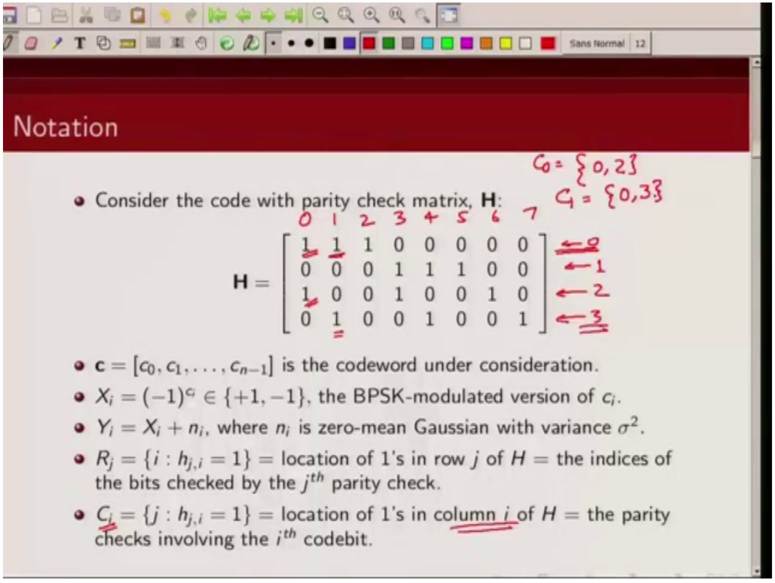
- Consider the code with parity check matrix, **H**:

$$\mathbf{H} = \begin{array}{ccccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & \leftarrow 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & \leftarrow 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & \leftarrow 2 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & \leftarrow 3 \end{array}$$

- $\mathbf{c} = [c_0, c_1, \ldots, c_{n-1}]$ 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 σ^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^{th} parity check.
- $C_i = \{j : h_{j,i} = 1\}$ = location of 1's in $\underline{\text{column } i}$ of H = the parity checks involving the i^{th} codebit.

Similarly c 1 is going to be 0, because it is participating in 0th parity check equation and 3, Ok. You can

(Refer Slide Time 12:56)



Notation

$C_0 = \{0, 2\}$

$C_1 = \{0, 3\}$

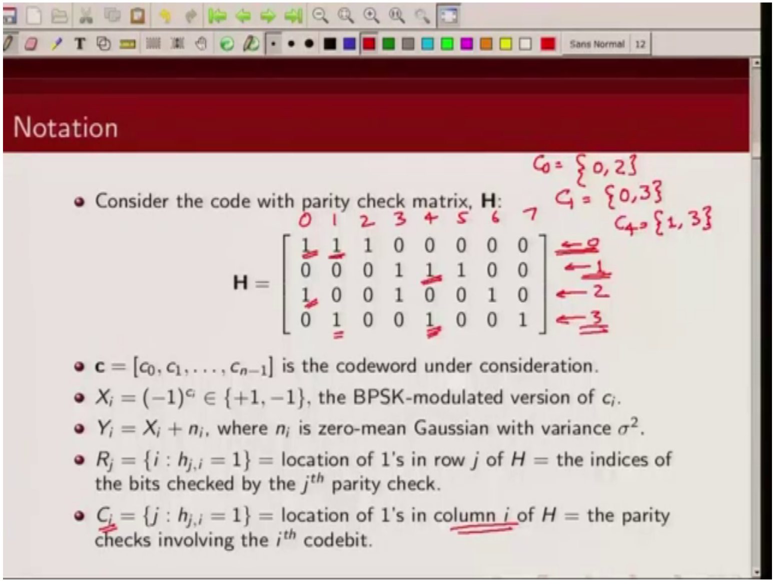
- Consider the code with parity check matrix, **H**:

$$\mathbf{H} = \begin{array}{c|cccccccc|c} & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & \\ \hline & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & \leftarrow 0 \\ & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & \leftarrow 1 \\ & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & \leftarrow 2 \\ & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & \leftarrow 3 \end{array}$$

- $\mathbf{c} = [c_0, c_1, \ldots, c_{n-1}]$ 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 σ^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^{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^{th} codebit.

take any, let's say c 4. What is c 4? Fourth bit is participating in parity check equation and participating in parity check equation 3. So

(Refer Slide Time 13:09)



Notation

$C_0 = \{0, 2\}$

$C_1 = \{0, 3\}$

$C_4 = \{1, 3\}$

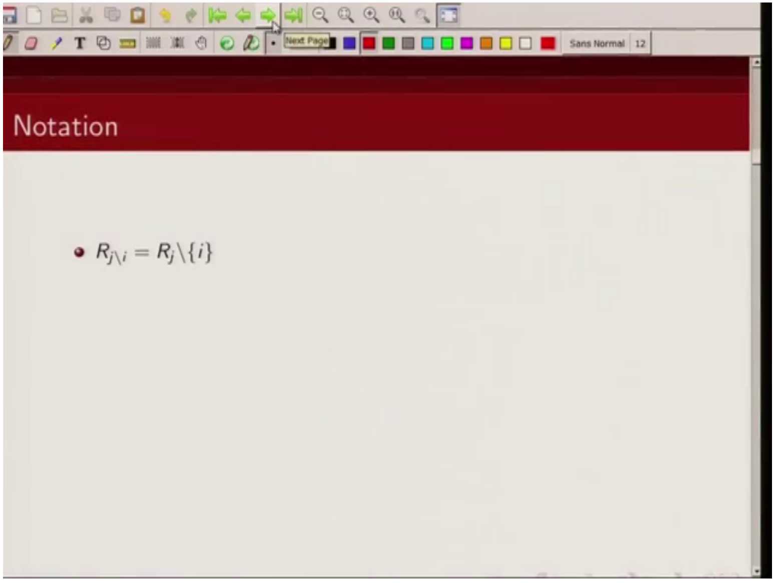
- Consider the code with parity check matrix, **H**:

$$\mathbf{H} = \begin{array}{c|cccccccc|c} & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & \\ \hline & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & \leftarrow 0 \\ & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & \leftarrow 1 \\ & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & \leftarrow 2 \\ & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & \leftarrow 3 \end{array}$$

- $\mathbf{c} = [c_0, c_1, \ldots, c_{n-1}]$ 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 σ^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^{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^{th} codebit.

like that you can find out what is c i, fine. Now

(Refer Slide Time 13:15)

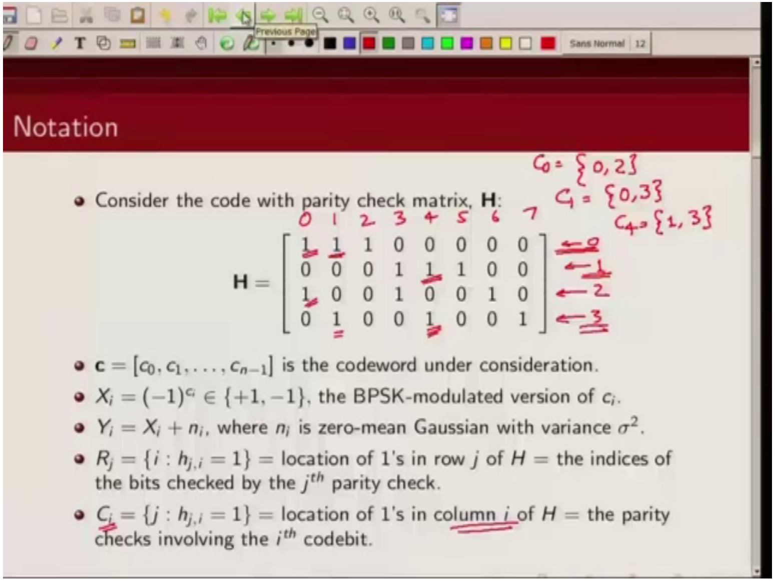


Notation

- $R_{j\backslash i} = R_j \backslash \{i\}$

we define the set R j minus a particular element i. So let's go

(Refer Slide Time 13:24)



Notation

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

$C_0 = \{0,2\}$

$C_1 = \{0,3\}$

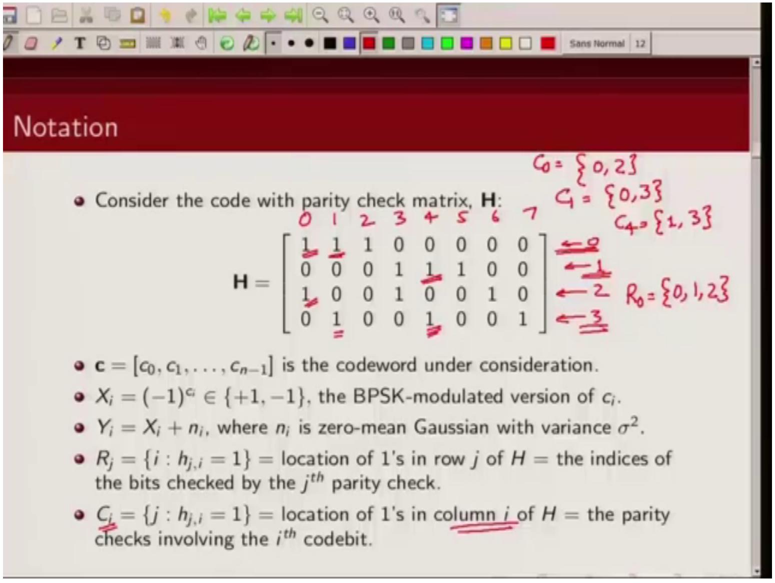
$C_4 = \{1,3\}$

$$\mathbf{H} = \begin{array}{cccccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & & \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & \leftarrow & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & \leftarrow & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & \leftarrow & 2 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & \leftarrow & 3 \end{array}$$

- $\mathbf{c} = [c_0, c_1, \ldots, c_{n-1}]$ 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 σ^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^{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^{th} codebit.

back. So what was our R, let's say R 0. What is R 0? R 0 was location of 1s in 0th row. So that location was 0,1 and 2

(Refer Slide Time 13:43)



Notation

$C_0 = \{0, 2\}$

$C_1 = \{0, 3\}$

$C_4 = \{1, 3\}$

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

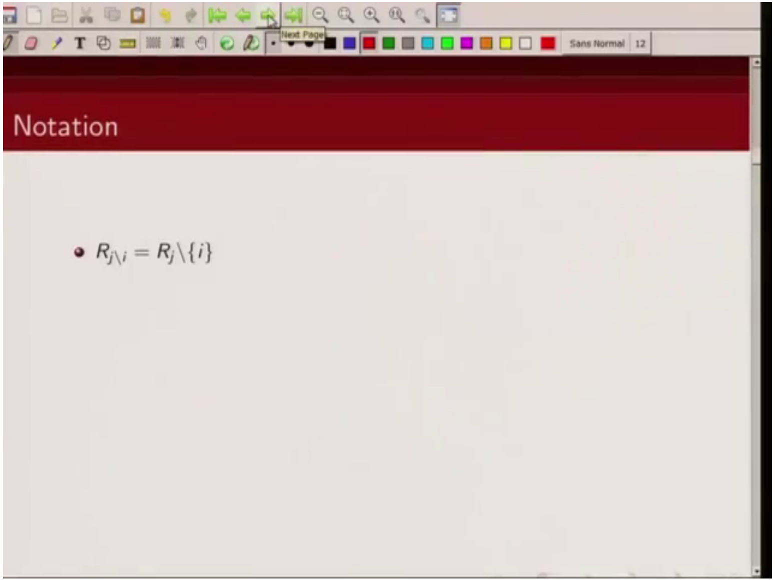
$$\mathbf{H} = \begin{array}{c|cccccccc|c} & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & \\ \hline & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & \leftarrow 0 \\ & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & \leftarrow 1 \\ & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & \leftarrow 2 \\ & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & \leftarrow 3 \end{array}$$

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

- $\mathbf{c} = [c_0, c_1, \ldots, c_{n-1}]$ 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 σ^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^{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^{th} codebit.

right? So if I define

(Refer Slide Time 13:47)

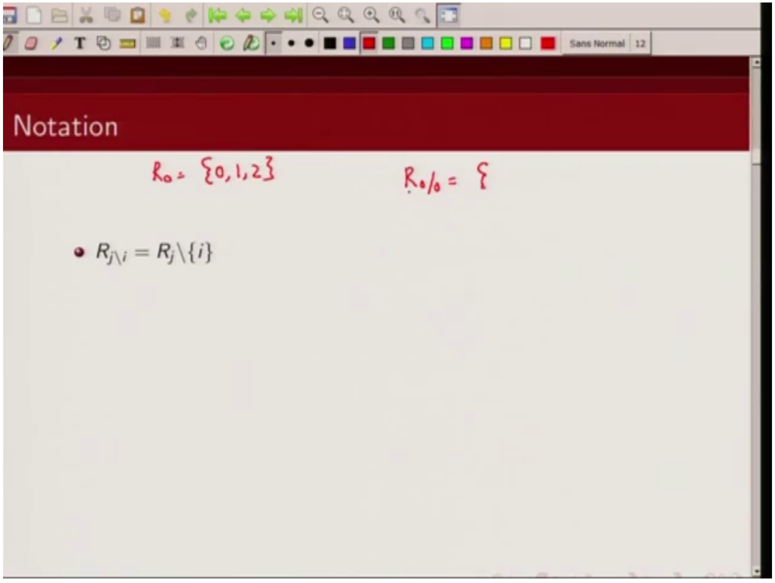


Notation

- $R_{j \setminus i} = R_j \setminus \{i\}$

R 0 slash 0, that would be then because R 0 R 0 is what, R 0 is 0,1, 2. So R 0

(Refer Slide Time 13:58)



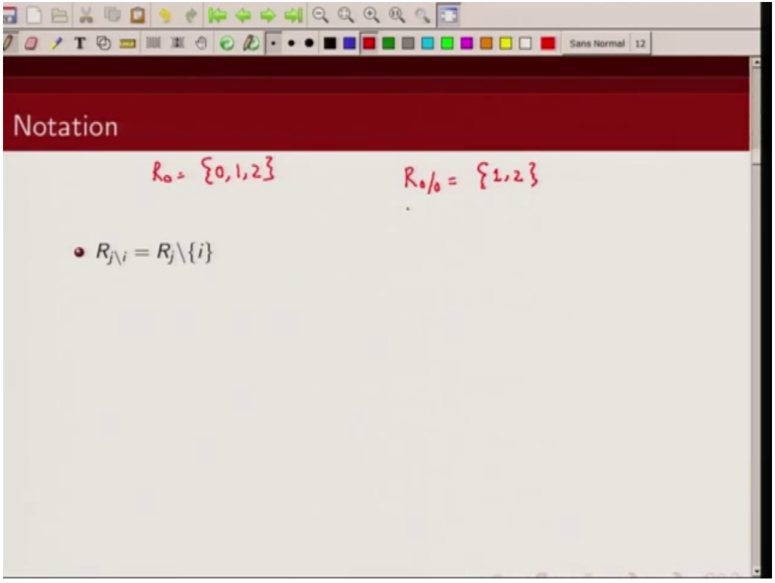
Notation

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

- $R_{j\setminus i} = R_j\setminus\{i\}$

minus 0 will have 1 and 2.

(Refer Slide Time 14:02)



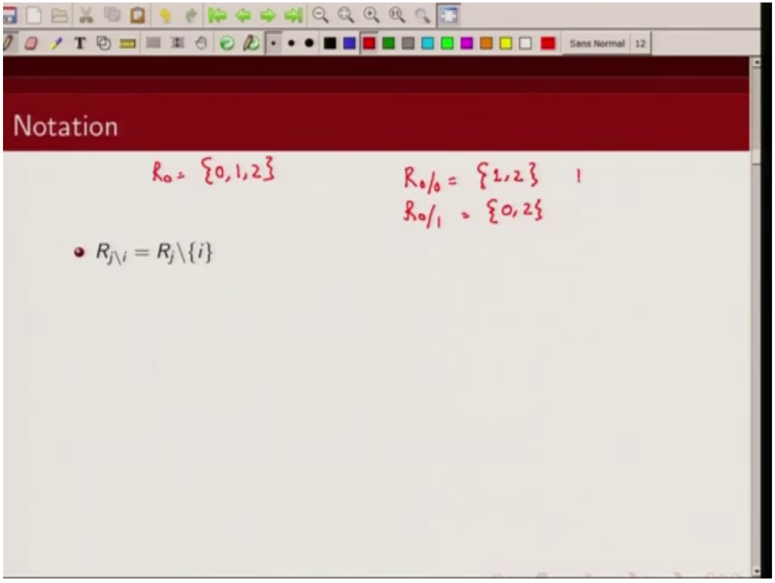
Notation

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

- $R_{j\setminus i} = R_j\setminus\{i\}$

Similarly R 0 minus 1, this will be set containing 0 and 2.

(Refer Slide Time 14:09)



Notation

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

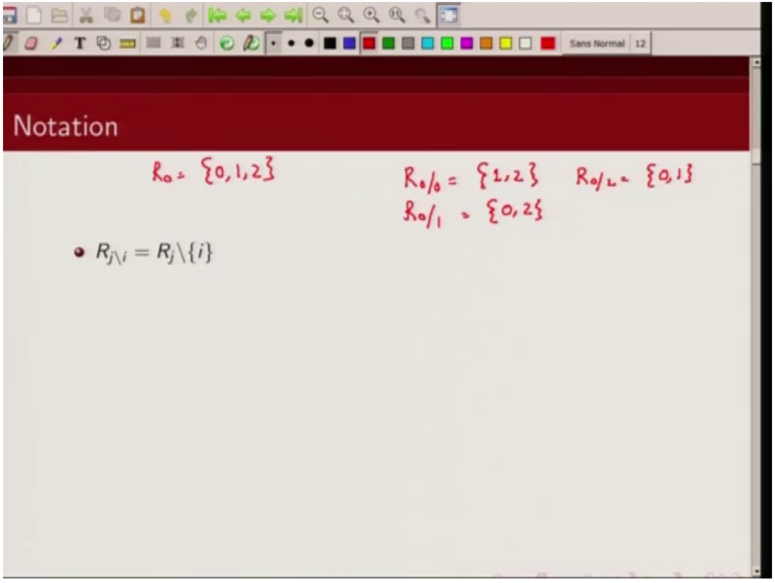
$R_{0/0} = \{1,2\}$!

$R_{0/1} = \{0,2\}$

- $R_{j\setminus i} = R_j\setminus\{i\}$

And this will be set containing 0 and 1.

(Refer Slide Time 14:14)



Notation

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

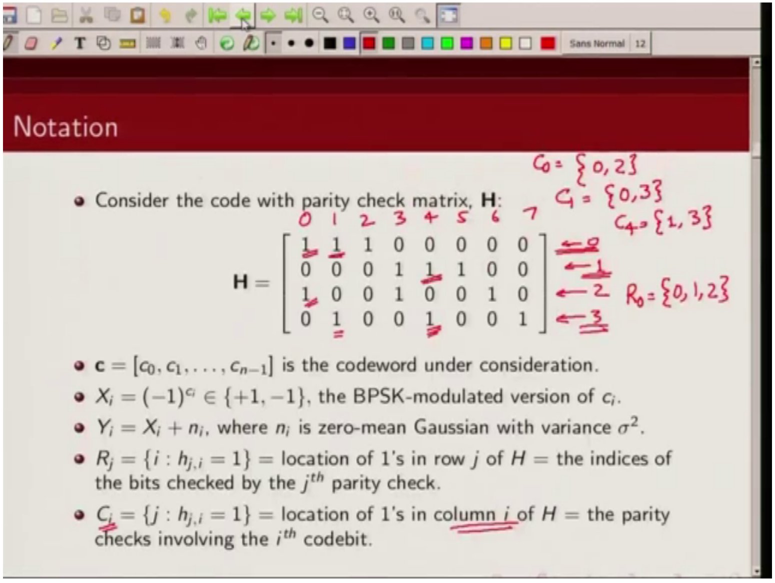
$R_{0/0} = \{1,2\}$ $R_{0/2} = \{0,1\}$

$R_{0/1} = \{0,2\}$

- $R_{j\setminus i} = R_j\setminus\{i\}$

Similarly we define this set, c i minus this element j, so for example,

(Refer Slide Time 14:26)



Notation

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

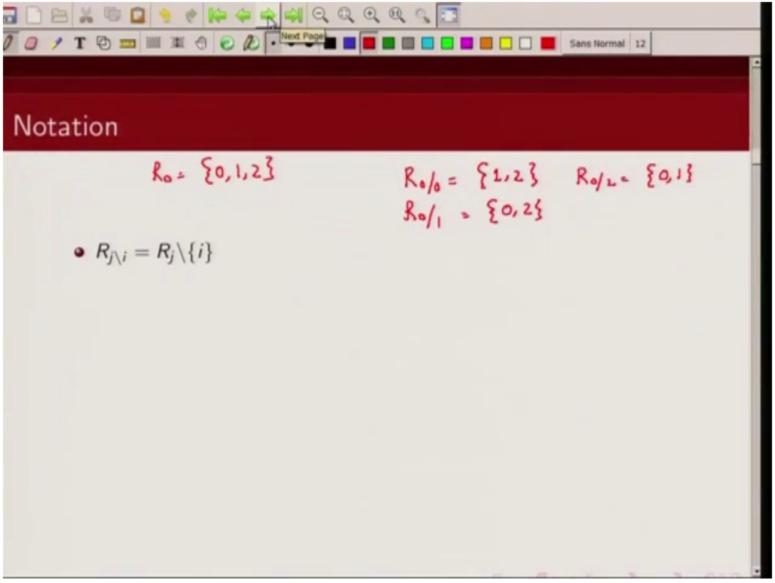
- Consider the code with parity check matrix, **H**:

$$\mathbf{H} = \begin{array}{cccccccc|l} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & \\ \hline \underline{1} & \underline{1} & 1 & 0 & 0 & 0 & 0 & 0 & \leftarrow \underline{\underline{0}} \\ 0 & 0 & 0 & 1 & \underline{1} & 1 & 0 & 0 & \leftarrow \underline{\underline{1}} \\ \underline{1} & 0 & 0 & 1 & 0 & 0 & 1 & 0 & \leftarrow 2 \quad R_0 = \{0,1,2\} \\ 0 & \underline{\underline{1}} & 0 & 0 & \underline{1} & 0 & 0 & 1 & \leftarrow \underline{\underline{3}} \end{array}$$

- $\mathbf{c} = [c_0, c_1, \ldots, c_{n-1}]$ 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 σ^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^{th} parity check.
- $\underline{\underline{C_i}} = \{j : h_{j,i} = 1\}$ = location of 1's in $\underline{\underline{\text{column } i}}$ of H = the parity checks involving the i^{th} codebit.

in this case let’s say c 1 is 0 3, so

(Refer Slide Time 14:31)



Notation

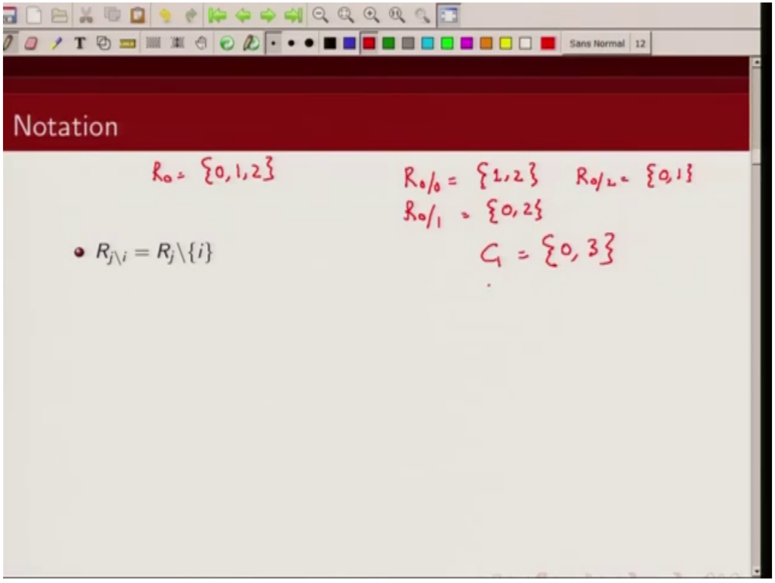
$R_0 = \{0,1,2\}$

$R_{0/0} = \{1,2\}$ $R_{0/2} = \{0,1\}$
$R_{0/1} = \{0,2\}$

- $R_{j\backslash i} = R_j \backslash \{i\}$

c 1 has elements 0 and 3. If we define

(Refer Slide Time 14:36)



Notation

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

- $R_{j\setminus i} = R_j\setminus\{i\}$

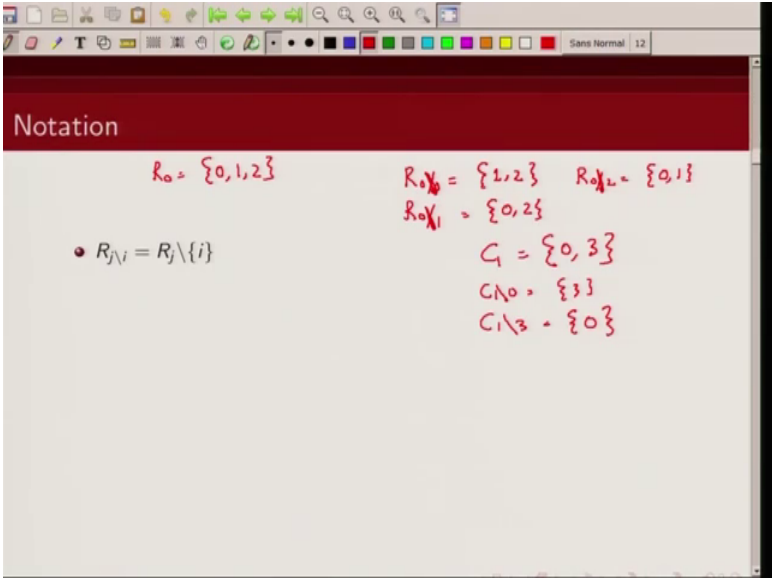
$R_{0/0} = \{1,2\}$ $R_{0/2} = \{0,1\}$

$R_{0/1} = \{0,2\}$

$C_1 = \{0,3\}$

this, this notation is like this, this zero, this will be 3 or c 1 0, Ok.

(Refer Slide Time 14:57)



Notation

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

- $R_{j\setminus i} = R_j\setminus\{i\}$

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

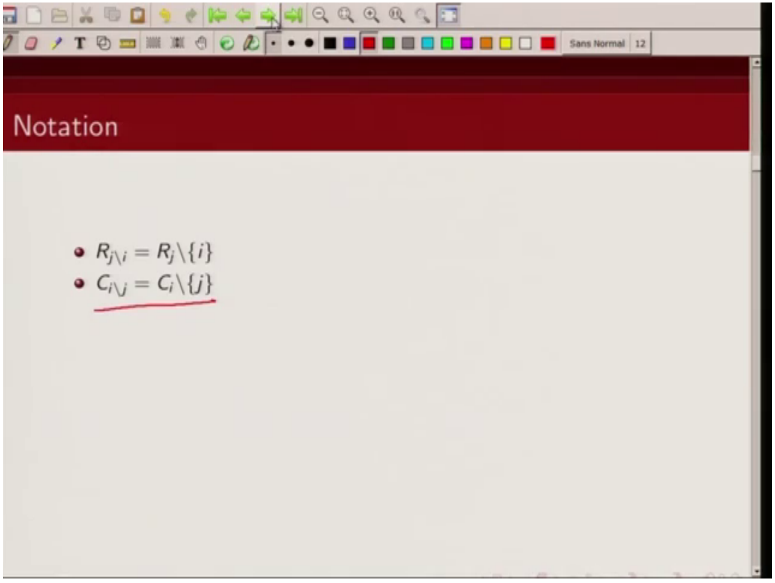
$R_{0\setminus 1} = \{0,2\}$

$C_1 = \{0,3\}$

$C\setminus 0 = \{3\}$

$C_1\setminus 3 = \{0\}$

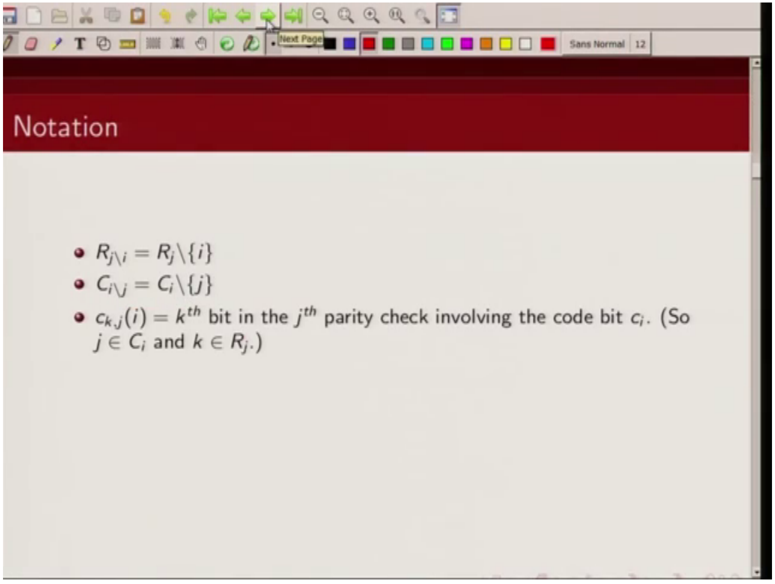
(Refer Slide Time 14:58)



Notation

- $R_{j\setminus i} = R_j \setminus \{i\}$
- $\underline{C_{i\setminus j} = C_i \setminus \{j\}}$

(Refer Slide Time 14:59)

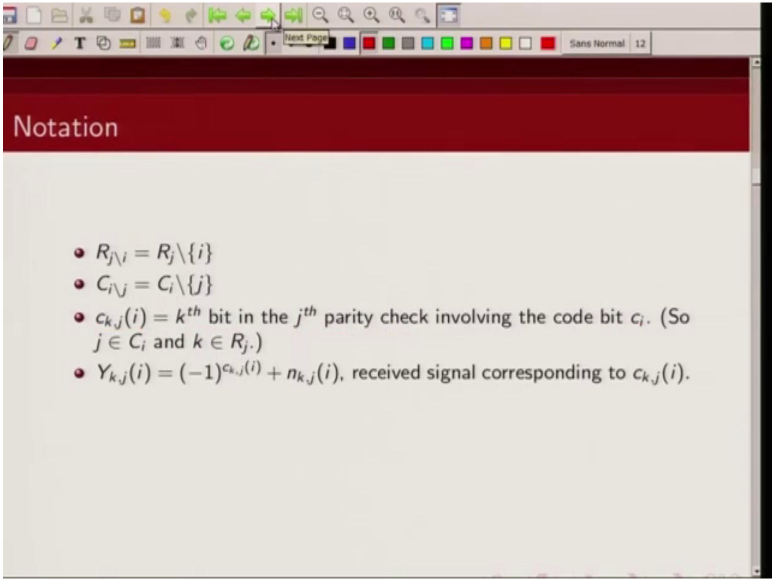


Notation

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

Now we define by c k j the kth bit in the jth parity check equation involving codebit c i. That is denoted by c i. c k j i.

(Refer Slide Time 15:16)

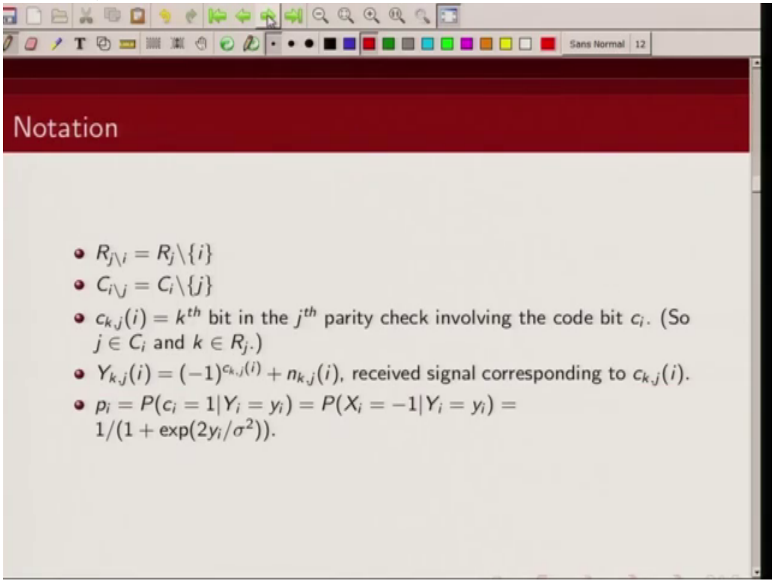


Notation

- $R_{j\setminus i} = R_j \setminus \{i\}$
- $C_{i\setminus j} = C_i \setminus \{j\}$
- $c_{k,j}(i) = k^{th}$ 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 $c_{k,j}(i)$.

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

(Refer Slide Time 15:31)

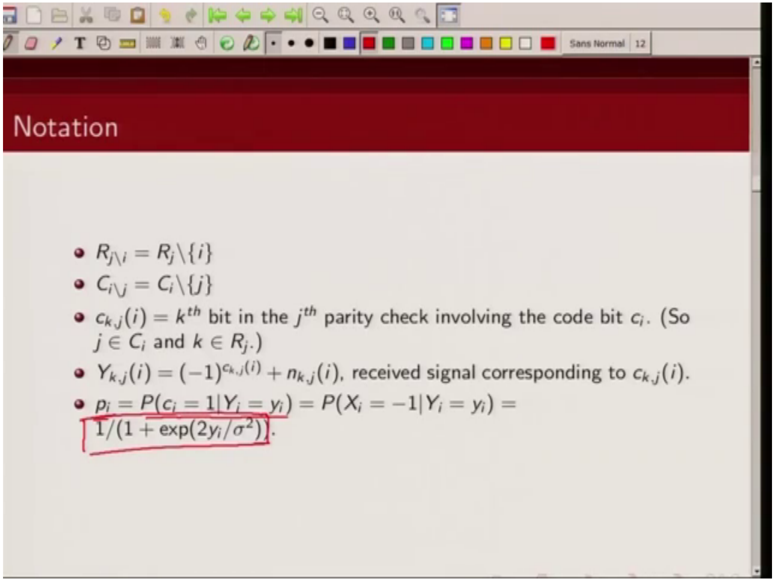


Notation

- $R_{j\setminus i} = R_j \setminus \{i\}$
- $C_{i\setminus j} = C_i \setminus \{j\}$
- $c_{k,j}(i) = k^{th}$ 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 $c_{k,j}(i)$.
- $p_i = P(c_i = 1|Y_i = y_i) = P(X_i = -1|Y_i = y_i) = 1/(1 + \exp(2y_i/\sigma^2))$.

Now for an a w g n channel we can compute this probability that c i b is 1 given a y i, this can be given by this expression.

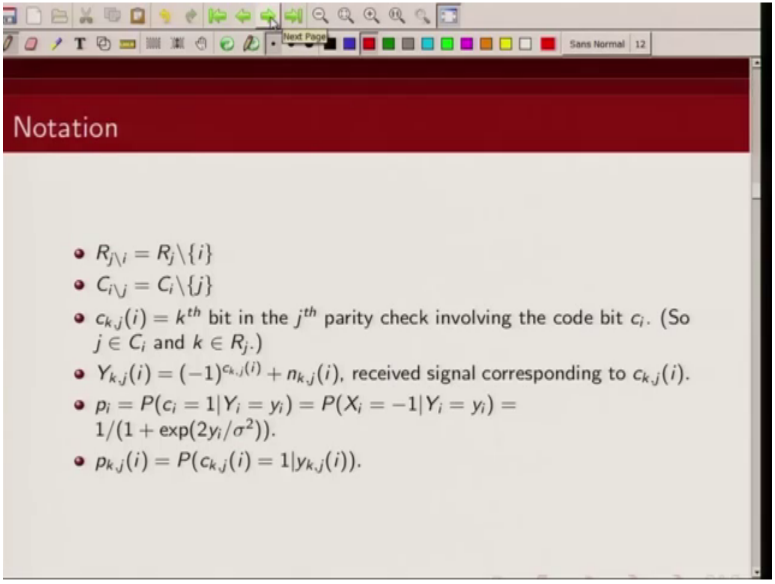
(Refer Slide Time 15:45)



Notation

- $R_{j\setminus i} = R_j \setminus \{i\}$
- $C_{i\setminus j} = C_i \setminus \{j\}$
- $c_{k,j}(i) = k^{th}$ 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 $c_{k,j}(i)$.
- $\underline{p_i = P(c_i = 1 | Y_i = y_i)} = P(X_i = -1 | Y_i = y_i) = \boxed{1/(1 + \exp(2y_i/\sigma^2))}$.

(Refer Slide Time 15:49)

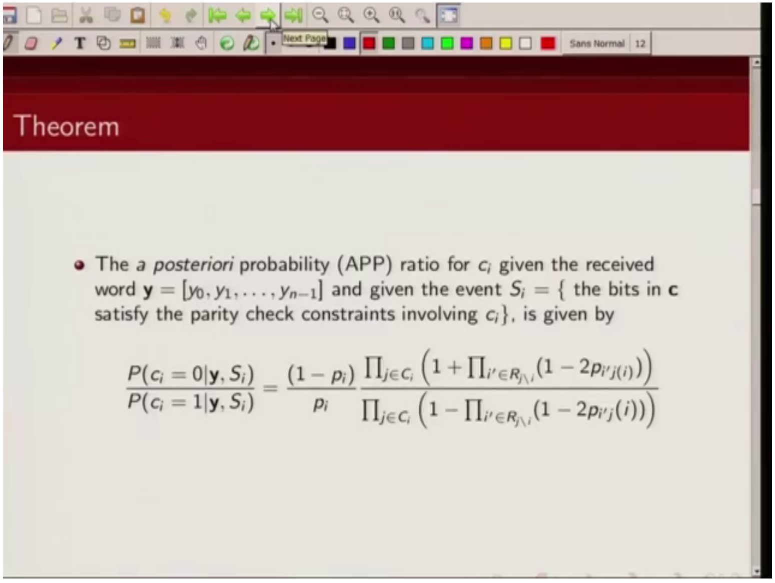


Notation

- $R_{j\setminus i} = R_j \setminus \{i\}$
- $C_{i\setminus j} = C_i \setminus \{j\}$
- $c_{k,j}(i) = k^{th}$ 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 $c_{k,j}(i)$.
- $p_i = P(c_i = 1 | Y_i = y_i) = P(X_i = -1 | Y_i = y_i) = 1/(1 + \exp(2y_i/\sigma^2))$.
- $p_{k,j}(i) = P(c_{k,j}(i) = 1 | y_{k,j}(i))$.

And we denote by p k j the probability that c k j is one given a received sequence y k j.

(Refer Slide Time 16:00)



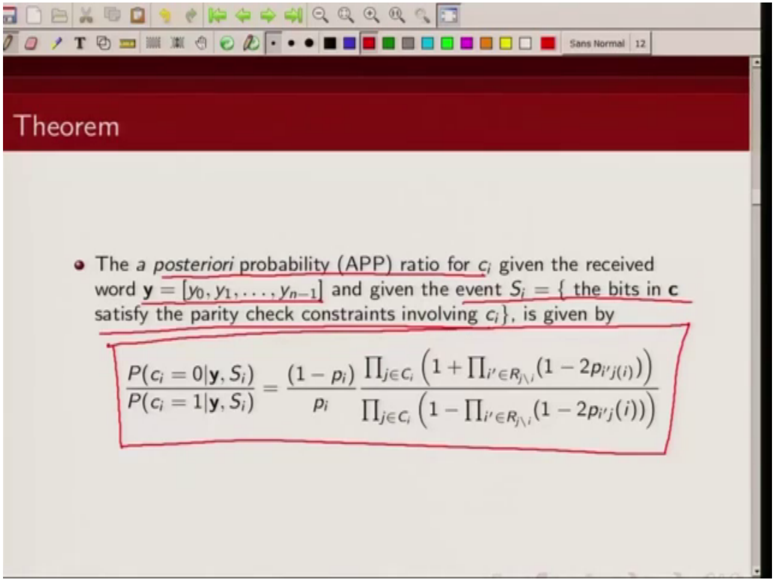
Theorem

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

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

So then we can write down the expression for a posteriori probability for our code bit c i given our received sequence is y and given that the parity check constraints containing c i has been satisfied. So what's the probability of c i being 0 given a received sequence y and given that the parity check constraint containing, involving c i has been satisfied. This is given by, divided by probability of c i being 1,given y and s i , this expression is given by this expression.

(Refer Slide Time 16:50)



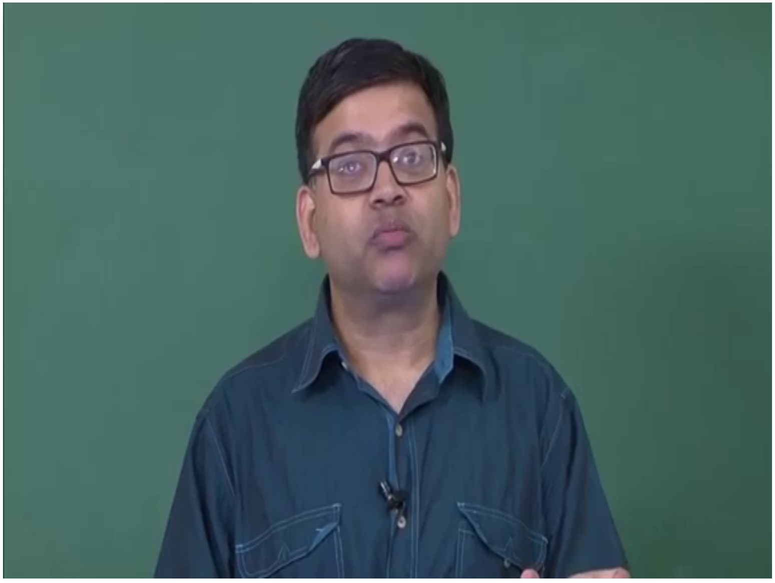
Theorem

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

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

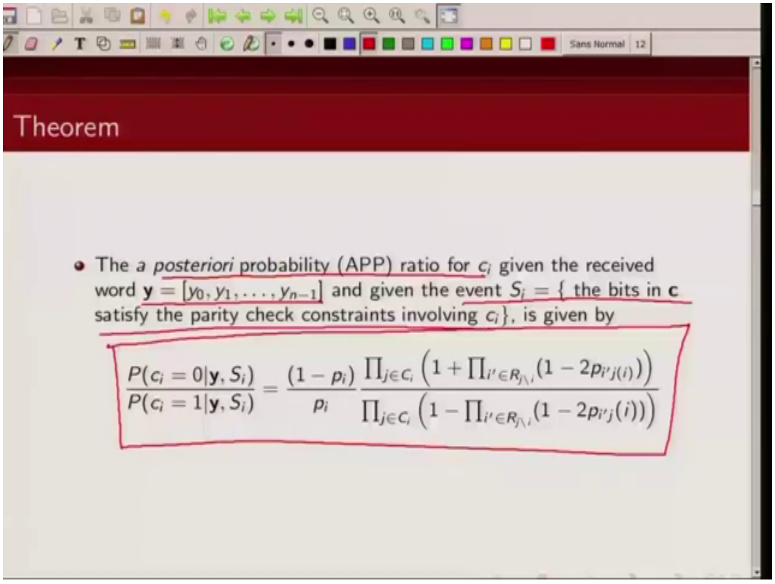
And we are going to use the theorem that we had proved in the beginning of the lecture to derive this expression, namely

(Refer Slide Time 17:01)



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 to derive

(Refer Slide Time 17:14)



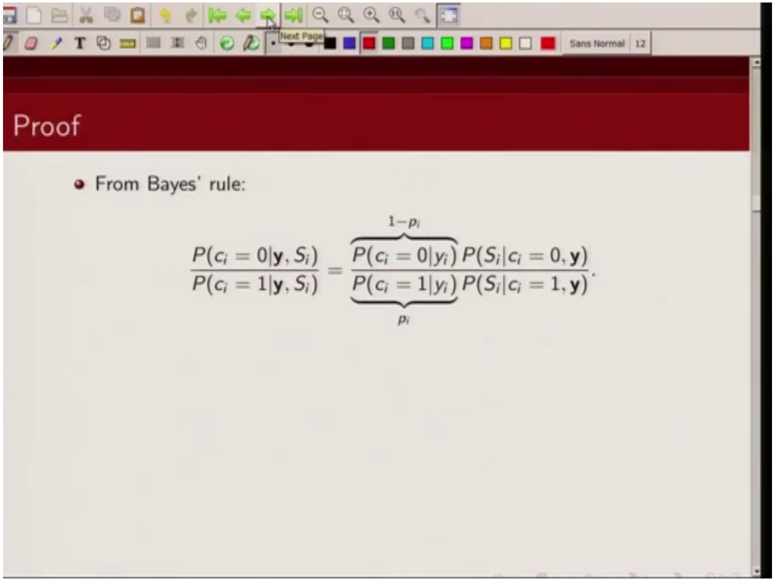
Theorem

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

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

this expression. So let's see.

(Refer Slide Time 17:18)



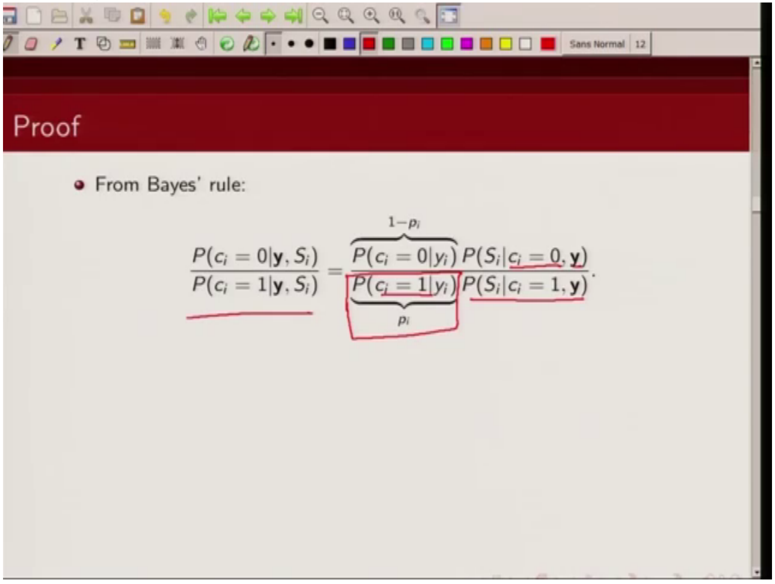
Proof

- From Bayes' rule:

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

So from Bayes' 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 zero and the received sequence is y. And similarly we can write that's probability of c i being 1 given received 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 18:00)



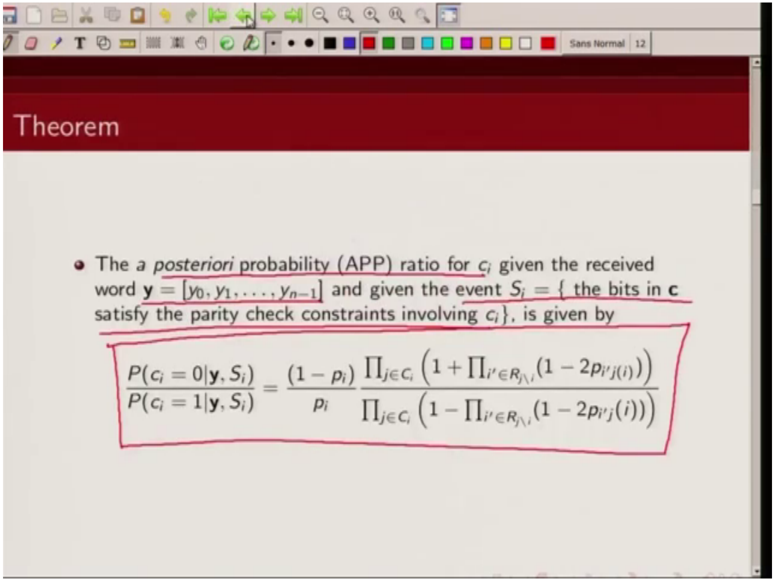
Proof

- From Bayes' rule:

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

If you go back

(Refer Slide Time 18:01)

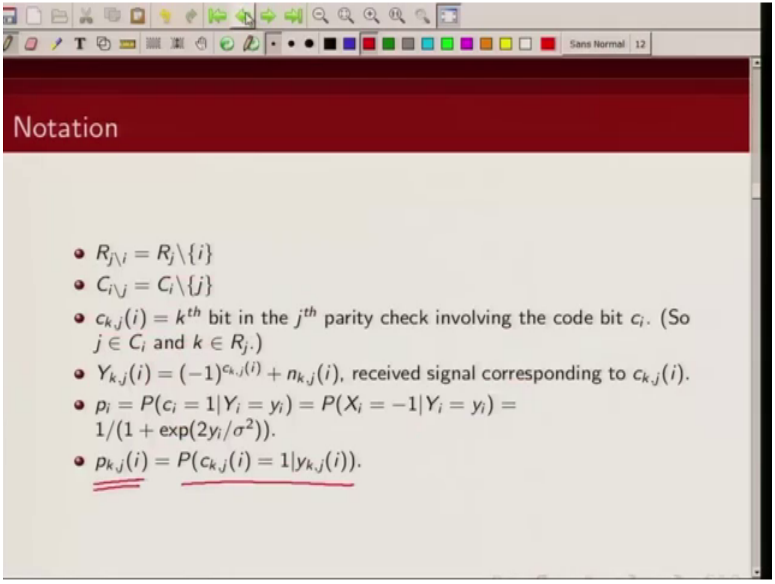


Theorem

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

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

(Refer Slide Time 18:02)

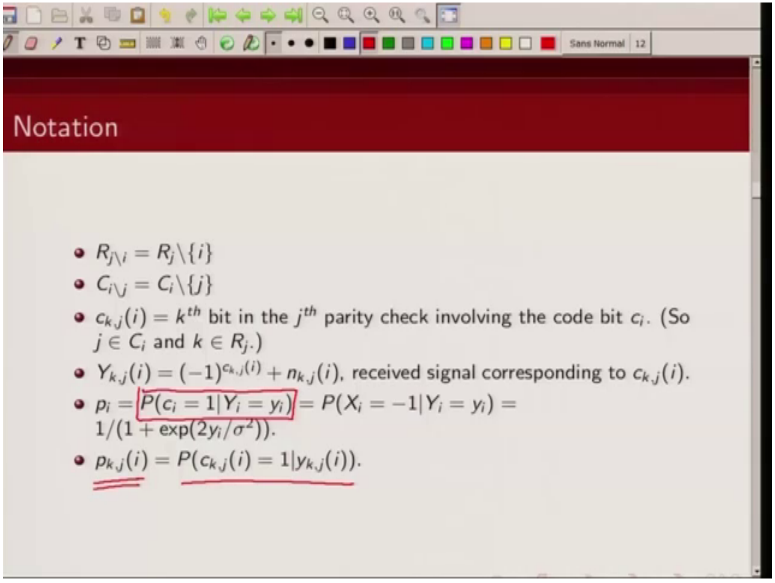


Notation

- $R_{j\setminus i} = R_j \setminus \{i\}$
- $C_{i\setminus j} = C_i \setminus \{j\}$
- $c_{k,j}(i) = k^{th}$ 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 $c_{k,j}(i)$.
- $p_i = P(c_i = 1 | Y_i = y_i) = P(X_i = -1 | Y_i = y_i) = 1/(1 + \exp(2y_i/\sigma^2))$.
- $p_{k,j}(i) = P(c_{k,j}(i) = 1 | y_{k,j}(i))$.

what was p i? p i is probability of c i given v i.

(Refer Slide Time 18:07)

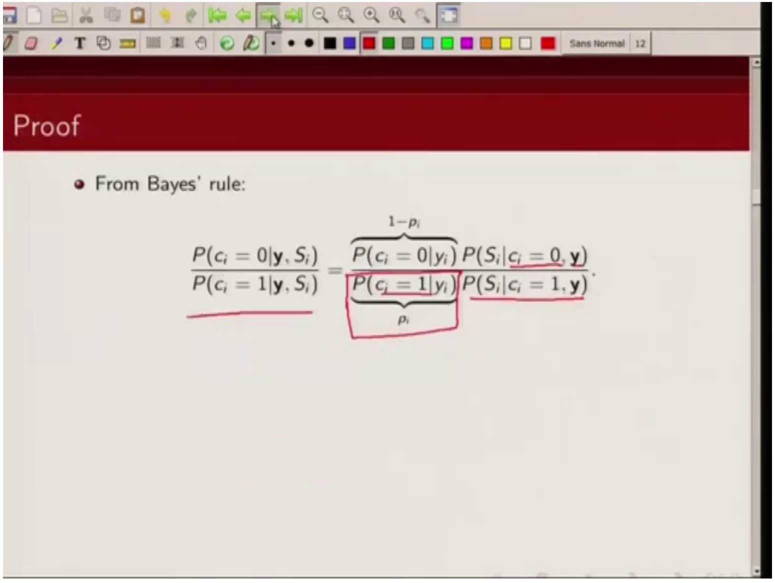


Notation

- $R_{j\backslash i} = R_j \backslash \{i\}$
- $C_{i\backslash j} = C_i \backslash \{j\}$
- $c_{k,j}(i) = k^{th}$ 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 $c_{k,j}(i)$.
- $p_i = \boxed{P(c_i = 1|Y_i = y_i)} = P(X_i = -1|Y_i = y_i) = 1/(1+\exp(2y_i/\sigma^2))$.
- $p_{k,j}(i) = P(c_{k,j}(i) = 1|y_{k,j}(i))$.

So that's p i.

(Refer Slide Time 18:09)



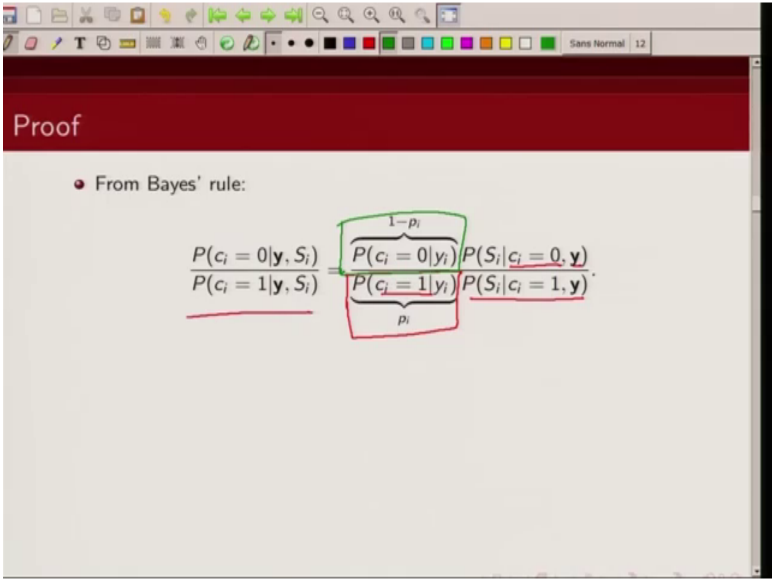
Proof

- From Bayes' rule:

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

So then this, upper term would be 1 minus p i, Ok.

(Refer Slide Time 18:15)



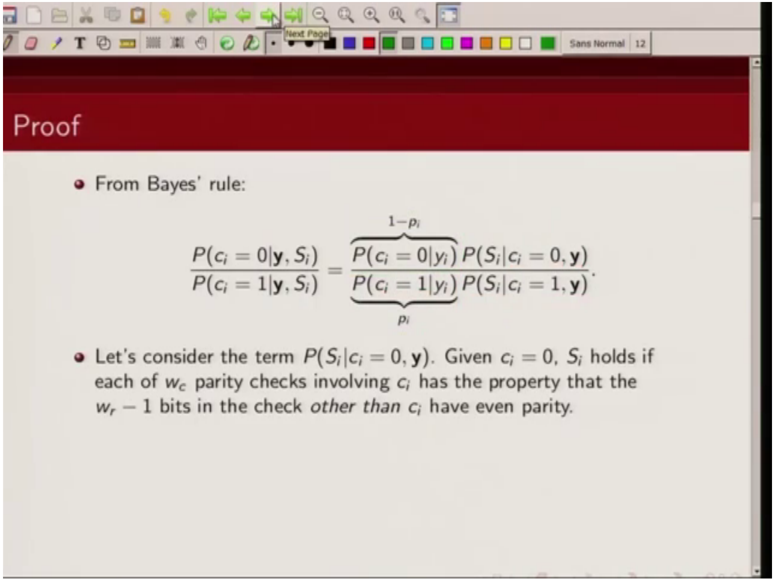
Proof

- From Bayes' rule:

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

Now let's pay close attention to these terms then.

(Refer Slide Time 18:22)



Proof

- From Bayes' rule:

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

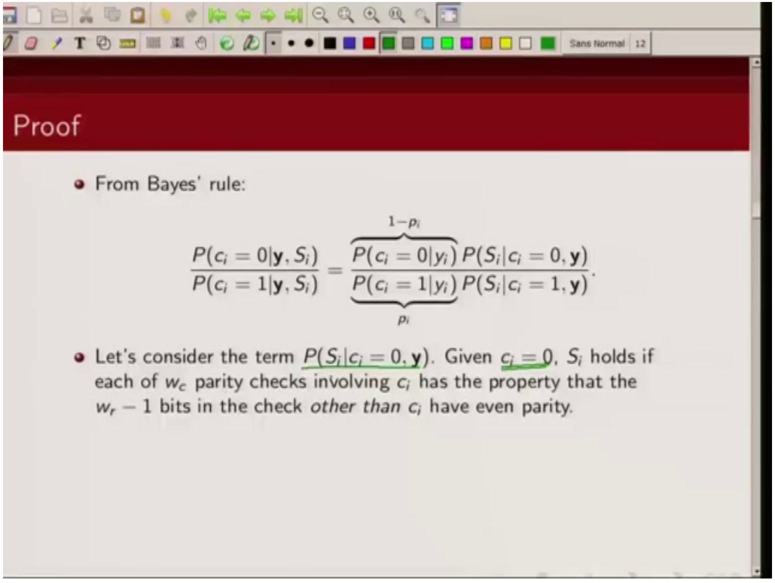
- Let's consider the term $P(S_i|c_i = 0, \mathbf{y})$. Given $c_i = 0$, S_i holds if each of w_c parity checks involving c_i has the property that the $w_r - 1$ bits in the check *other than* c_i have even parity.

So what is this? Given that my codebit c i is 0, when will parity check constraint involving c i will be satisfied? It is when sum of other parity bits

(Refer Slide Time 18:41)

involved in the parity check constraints, they add up to have even parity, right? So

(Refer Slide Time 18:49)



Proof

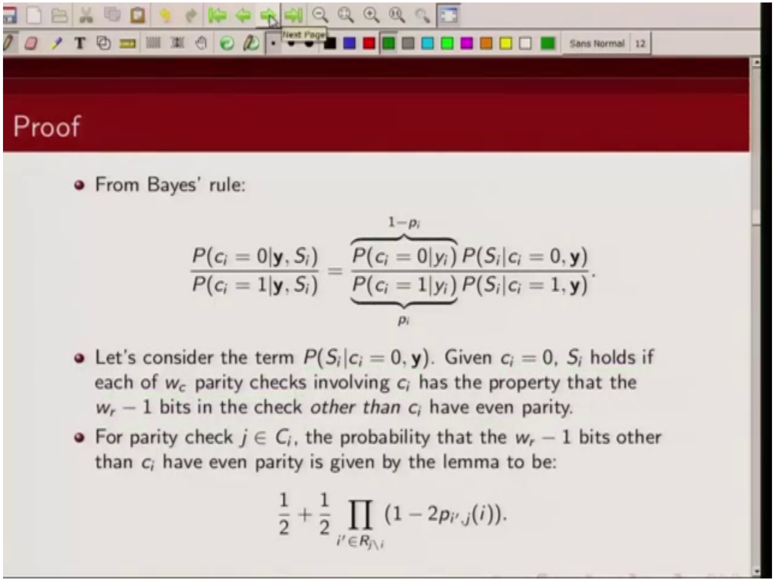
- From Bayes' rule:

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

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

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 a regular L D P C code that number is w 1 minus 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 19:53)



Proof

- From Bayes' rule:

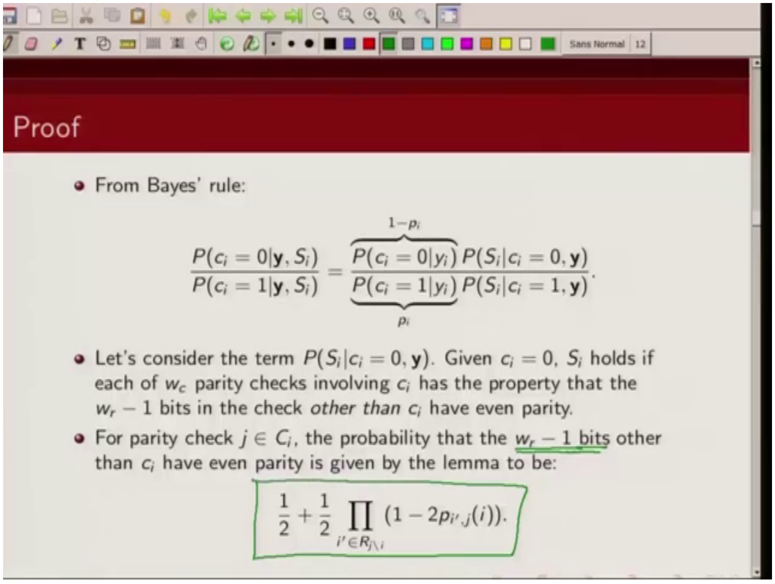
$$\frac{P(c_i=0|\mathbf{y},S_i)}{P(c_i=1|\mathbf{y},S_i)} = \frac{\overbrace{P(c_i=0|y_i)}^{1-p_i}\,P(S_i|c_i=0,\mathbf{y})}{\underbrace{P(c_i=1|y_i)}_{p_i}\,P(S_i|c_i=1,\mathbf{y})}.$$

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

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

Now from the theorem that we have proved, we know what is that probability. Probability that other than c i, w or minus bits, they have even parity, that probability is given by this expression.

(Refer Slide Time 20:13)



Proof

- From Bayes' rule:

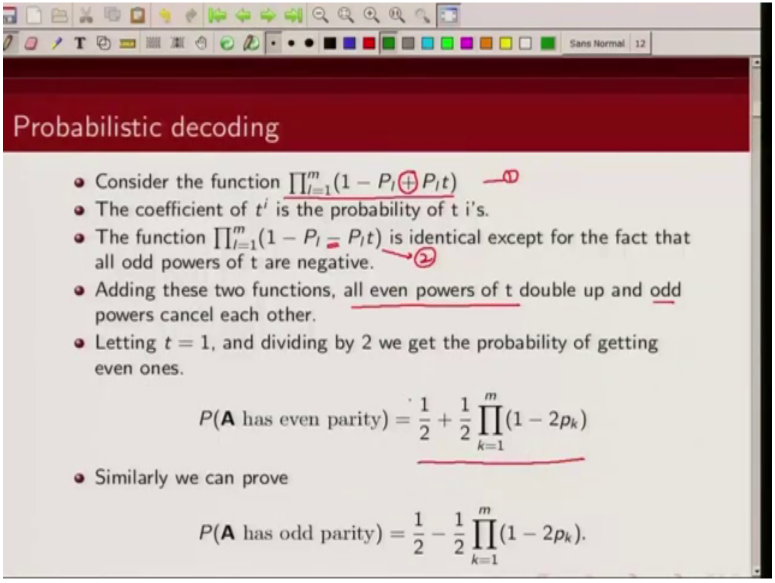
$$\frac{P(c_i=0|\mathbf{y},S_i)}{P(c_i=1|\mathbf{y},S_i)} = \frac{\overbrace{P(c_i=0|y_i)}^{1-p_i}\,P(S_i|c_i=0,\mathbf{y})}{\underbrace{P(c_i=1|y_i)}_{p_i}\,P(S_i|c_i=1,\mathbf{y})}.$$

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

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

If you go back probability

(Refer Slide Time 20:17)



Probabilistic decoding

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

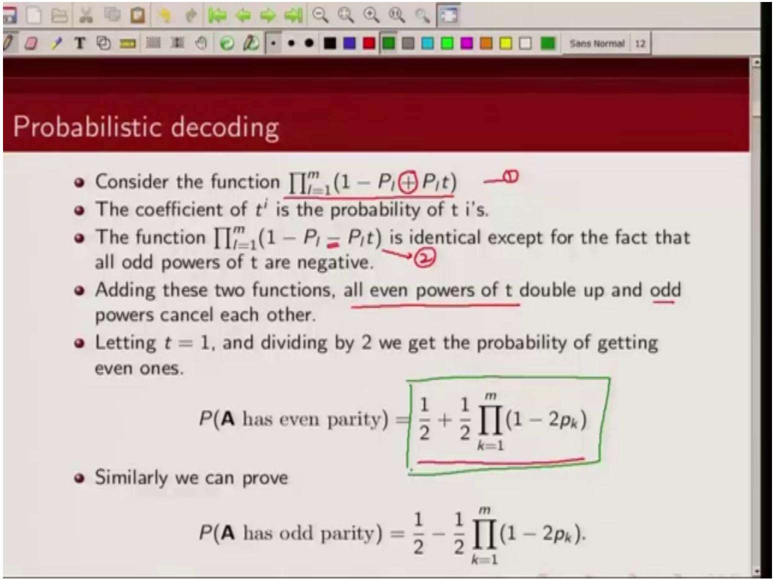
$$P(\mathbf{A} \text{ has even parity}) = \underline{\frac{1}{2} + \frac{1}{2}\prod_{k=1}^{m}(1-2p_k)}$$

- Similarly we can prove

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

that they have, probability

(Refer Slide Time 20:21)



Probabilistic decoding

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

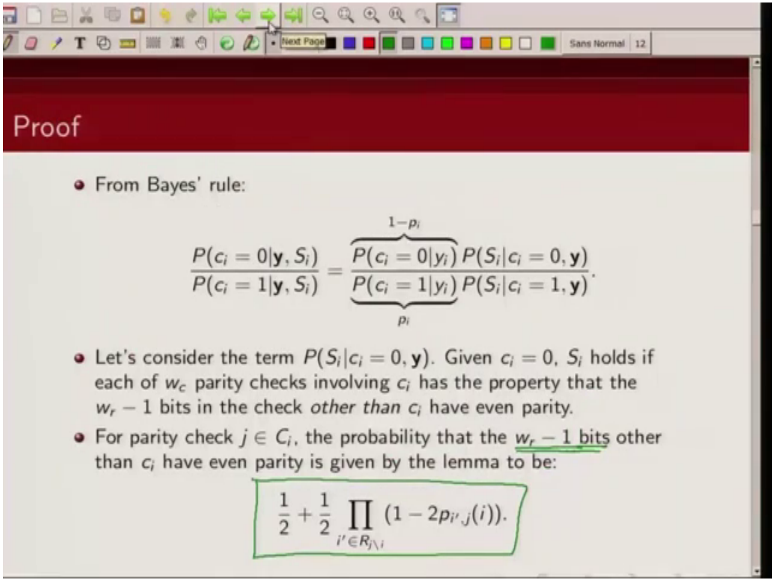
$$P(\mathbf{A} \text{ has even parity}) = \boxed{\underline{\frac{1}{2} + \frac{1}{2}\prod_{k=1}^{m}(1-2p_k)}}$$

- Similarly we can prove

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

that m random variables have even parity is half plus half product of 1 minus 2 p k. So probability that w r minus 1, so half

(Refer Slide Time 20:37)



Proof

- From Bayes' rule:

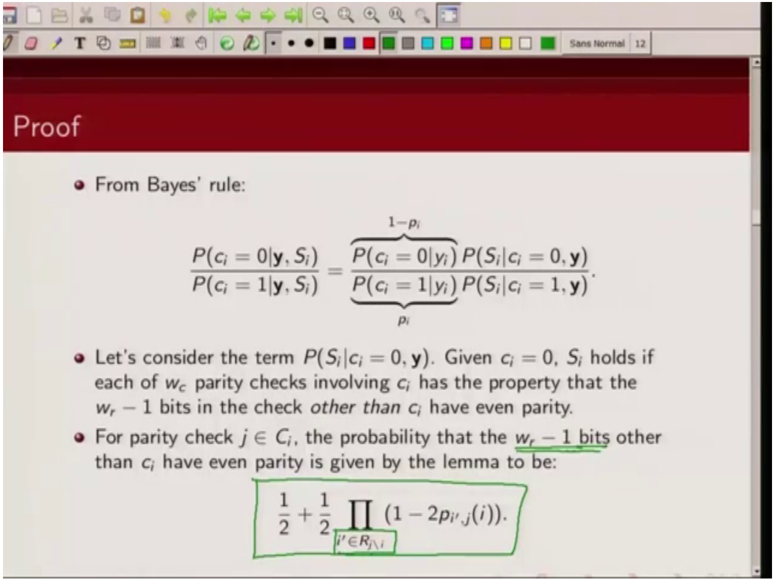
$$\frac{P(c_i=0|\mathbf{y},S_i)}{P(c_i=1|\mathbf{y},S_i)}=\frac{\overbrace{P(c_i=0|y_i)}^{1-p_i}P(S_i|c_i=0,\mathbf{y})}{\underbrace{P(c_i=1|y_i)}_{p_i}P(S_i|c_i=1,\mathbf{y})}.$$

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

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

plus half, now pay close attention to this.

(Refer Slide Time 20:42)



Proof

- From Bayes' rule:

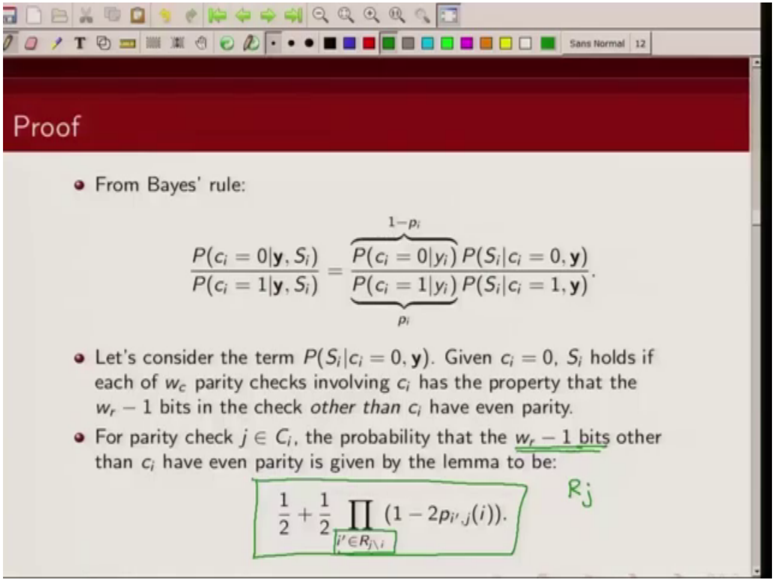
$$\frac{P(c_i=0|\mathbf{y},S_i)}{P(c_i=1|\mathbf{y},S_i)}=\frac{\overbrace{P(c_i=0|y_i)}^{1-p_i}P(S_i|c_i=0,\mathbf{y})}{\underbrace{P(c_i=1|y_i)}_{p_i}P(S_i|c_i=1,\mathbf{y})}.$$

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

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

This is 1 minus 2 p k. Now what are the bits that we are considering. Now what will R j tell us? R j

(Refer Slide Time 20:51)



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_c parity checks involving c_i has the property that the $w_r - 1$ bits in the check *other than* c_i have even parity.
- For parity check $j \in C_i$, the probability that the $w_r - 1$ bits other than c_i have even parity is given by the lemma to be:

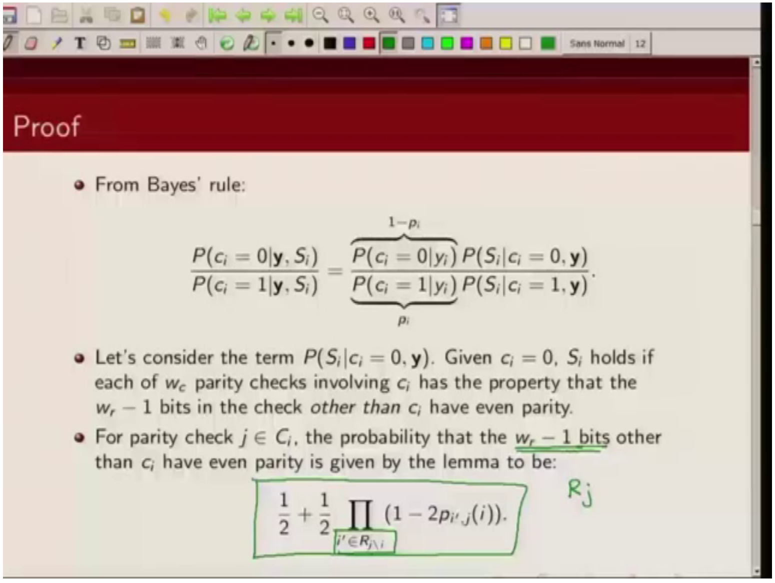
$$\boxed{\frac{1}{2} + \frac{1}{2} \prod_{\boxed{i' \in R_{j \setminus i}}} (1 - 2p_{i',j}(i)).} \quad R_j$$

will tell us

(Refer Slide Time 20:53)

jth parity check equation. And

(Refer Slide Time 20:56)



Proof

- From Bayes' rule:

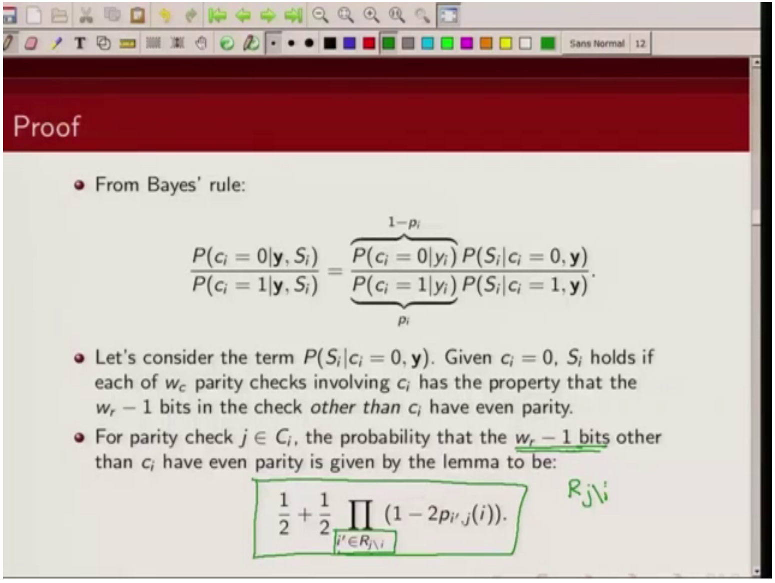
$$\frac{P(c_i = 0|\mathbf{y}, S_i)}{P(c_i = 1|\mathbf{y}, S_i)} = \frac{\overbrace{P(c_i = 0|y_i)}^{1-p_i}\, P(S_i|c_i = 0, \mathbf{y})}{\underbrace{P(c_i = 1|y_i)}_{p_i}\, P(S_i|c_i = 1, \mathbf{y})}.$$

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

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

R j minus i, where i is, c i bit is involved,

(Refer Slide Time 21:00)



Proof

- From Bayes' rule:

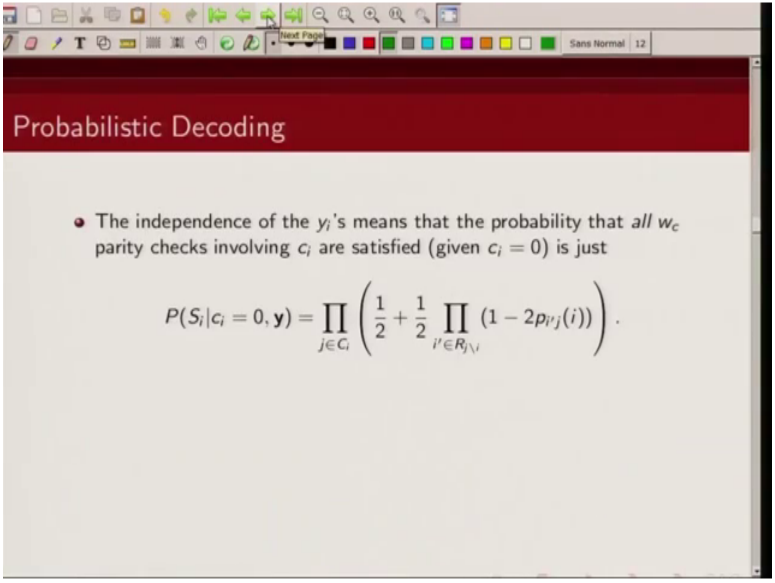
$$\frac{P(c_i = 0|\mathbf{y}, S_i)}{P(c_i = 1|\mathbf{y}, S_i)} = \frac{\overbrace{P(c_i = 0|y_i)}^{1-p_i}\, P(S_i|c_i = 0, \mathbf{y})}{\underbrace{P(c_i = 1|y_i)}_{p_i}\, P(S_i|c_i = 1, \mathbf{y})}.$$

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

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

and this is minus i, so the other bits, other than c i which are participating in the parity check constraint, product of this, they should add up to have even parity. So this is a set where c i is participating in a parity check constraint. So other than c i, other bits have even parity. That is captured by this particular set.

(Refer Slide Time 21:28)



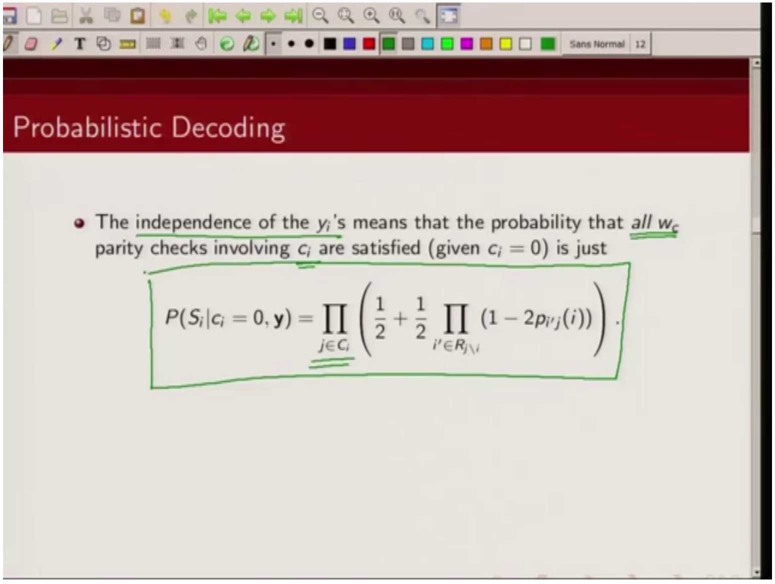
Probabilistic Decoding

- 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

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

And this should hold for all parity check equations involving c i. So this should hold for all w c parity check sets where this particular c i is participating. So that's why you assuming that c i's are independent I can then write the probability of product over all such parity check equations where this is involved. So I can write down then probability that my

(Refer Slide Time 22:06)



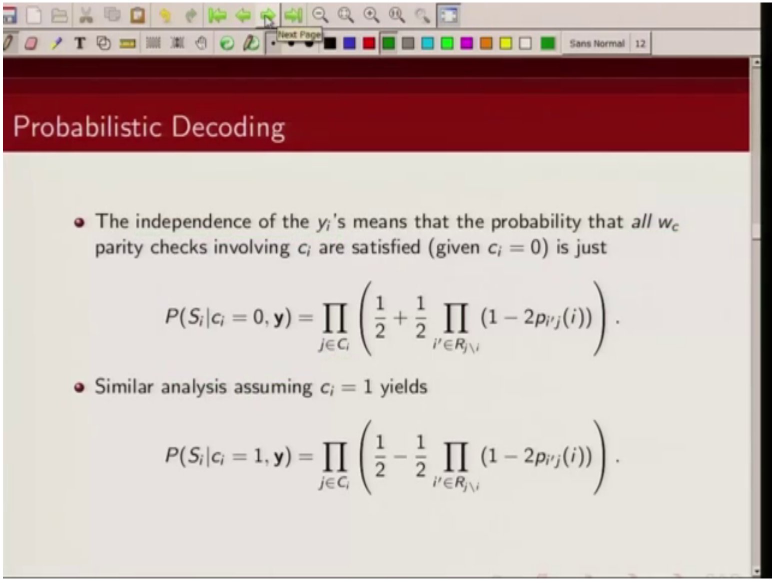
Probabilistic Decoding

- 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

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

parity check set constraint is satisfied given c i is 0, is given by this expression. And I can

(Refer Slide Time 22:17)



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

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

- Similar analysis assuming $c_i = 1$ yields

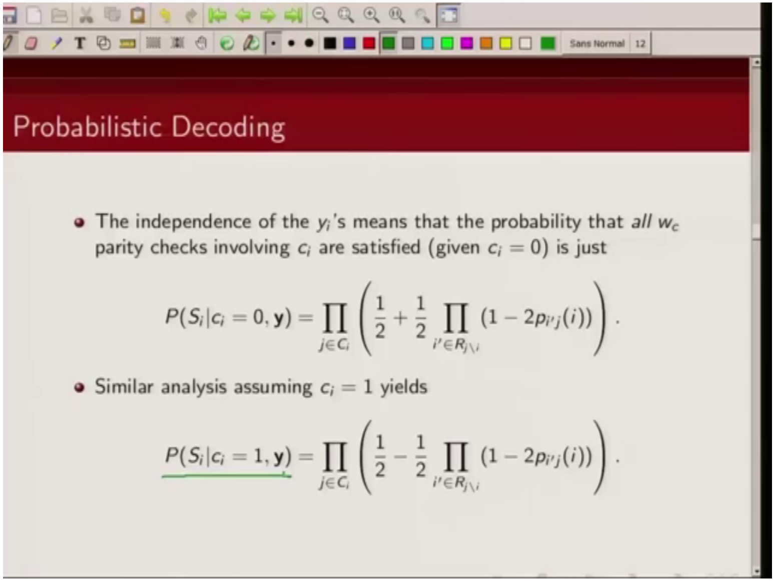
$$P(S_i|c_i = 1, \mathbf{y}) = \prod_{j \in C_i} \left(\frac{1}{2} - \frac{1}{2} \prod_{i' \in R_{j \setminus i}} (1 - 2p_{i'j}(i)) \right).$$

follow the same logic to find out the probability when c i is 1. When c i is 1, what you want, the other bits should add up to have

(Refer Slide Time 22:30)

an odd parity. And that is

(Refer Slide Time 22:32)



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

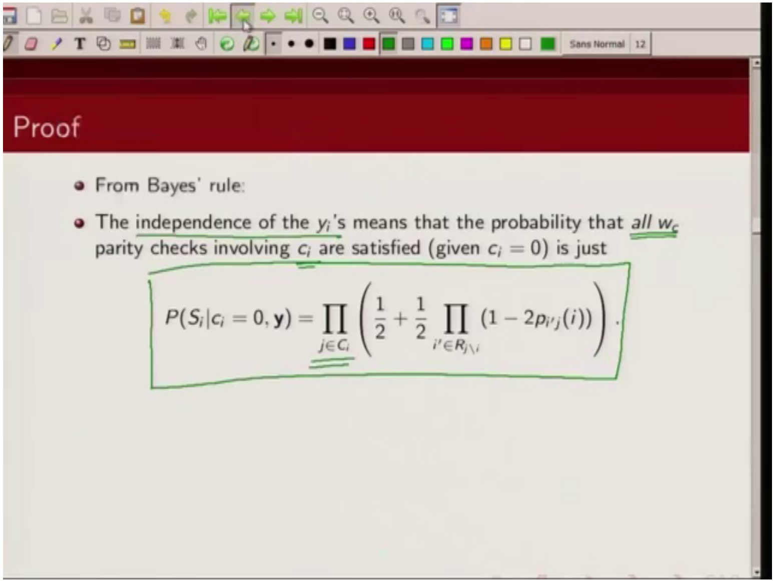
$$P(S_i|c_i = 0, \mathbf{y}) = \prod_{j \in C_i} \left(\frac{1}{2} + \frac{1}{2} \prod_{i' \in R_{j \setminus i}} (1 - 2p_{i'j}(i)) \right).$$

- Similar analysis assuming $c_i = 1$ yields

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

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 each of these probabilities. So this is a probability of this parity check set, I mean parity check constraint getting satisfied when c i is 1 and this is expression when c i is 0. So if plug these values

(Refer Slide Time 23:03)



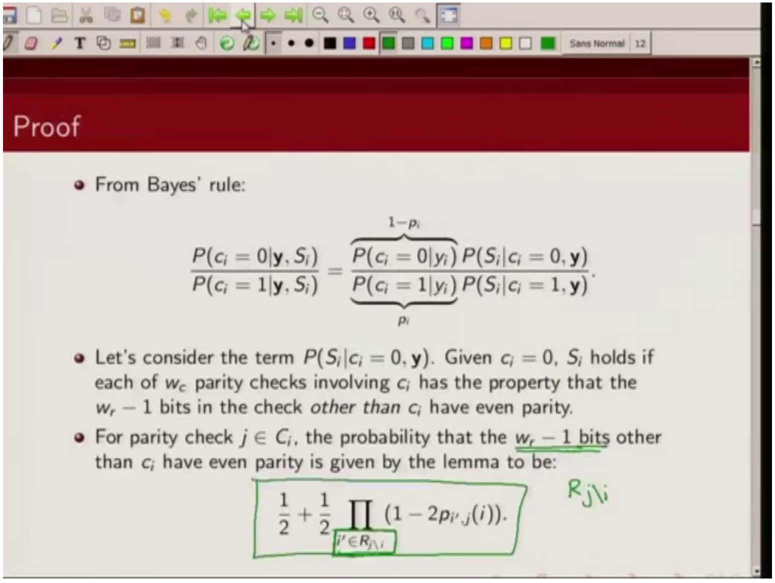
Proof

- From Bayes' rule:
- 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

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

in my expression

(Refer Slide Time 23:04)



Proof

- From Bayes' rule:

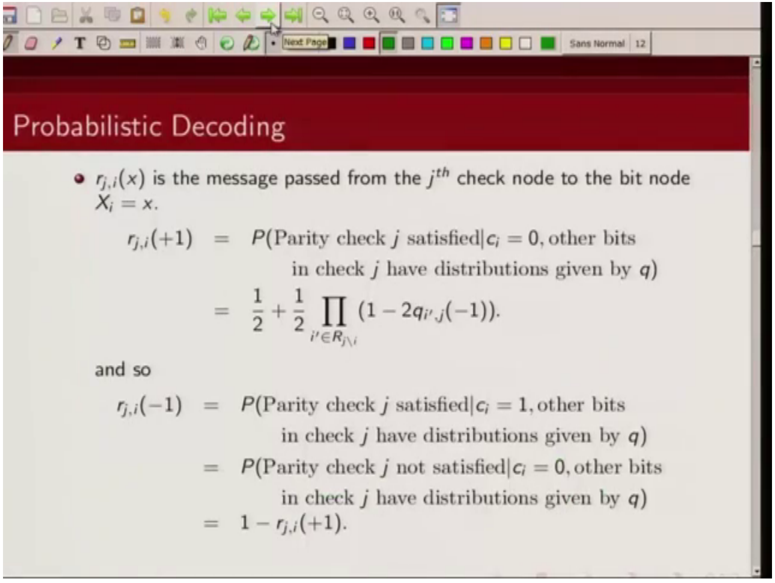
$$\frac{P(c_i = 0|\mathbf{y}, S_i)}{P(c_i = 1|\mathbf{y}, S_i)} = \frac{\overbrace{P(c_i = 0|y_i)}^{1-p_i} P(S_i|c_i = 0, \mathbf{y})}{\underbrace{P(c_i = 1|y_i)}_{p_i} P(S_i|c_i = 1, \mathbf{y})}.$$

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

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

here, this expression what I get is the

(Refer Slide Time 23:10)



Probabilistic Decoding

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

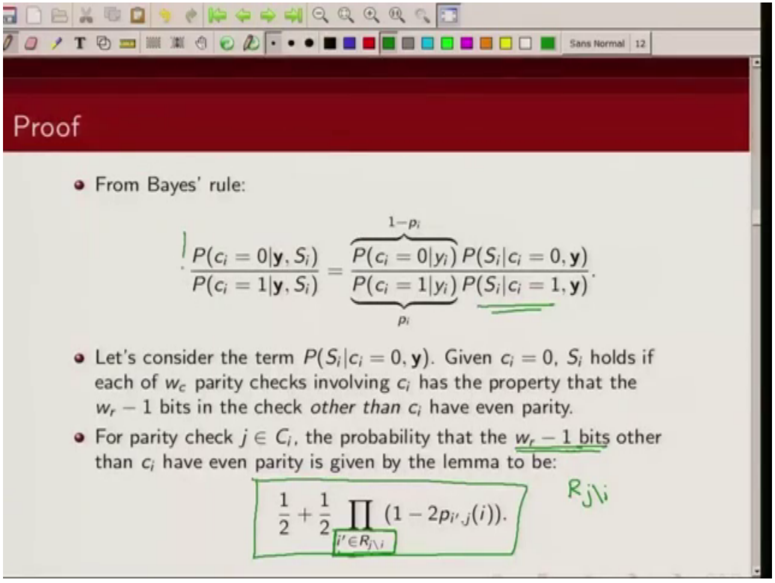
$$\begin{aligned} r_{j,i}(+1) &= P(\text{Parity check } j \text{ satisfied}|c_i = 0, \text{other bits} \\ &\qquad \text{in check } j \text{ have distributions given by } q) \\ &= \frac{1}{2} + \frac{1}{2} \prod_{i' \in R_{j \setminus i}} (1 - 2q_{i',j}(-1)). \end{aligned}$$

and so

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

the expression for

(Refer Slide Time 23:12)



Proof

- From Bayes' rule:

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

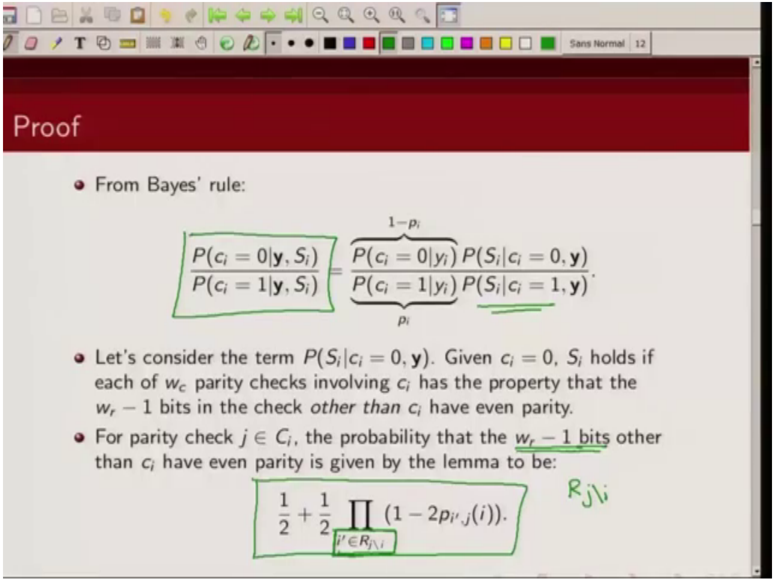
- Let's consider the term $P(S_i|c_i = 0, \mathbf{y})$. Given $c_i = 0$, S_i holds if each of w_c parity checks involving c_i has the property that the $w_r - 1$ bits in the check *other than* c_i have even parity.
- For parity check $j \in C_i$, the probability that the $\underline{w_r - 1 \text{ bits}}$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

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

$R_{j\setminus i}$

this, Ok.

(Refer Slide Time 23:14)



Proof

- From Bayes' rule:

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

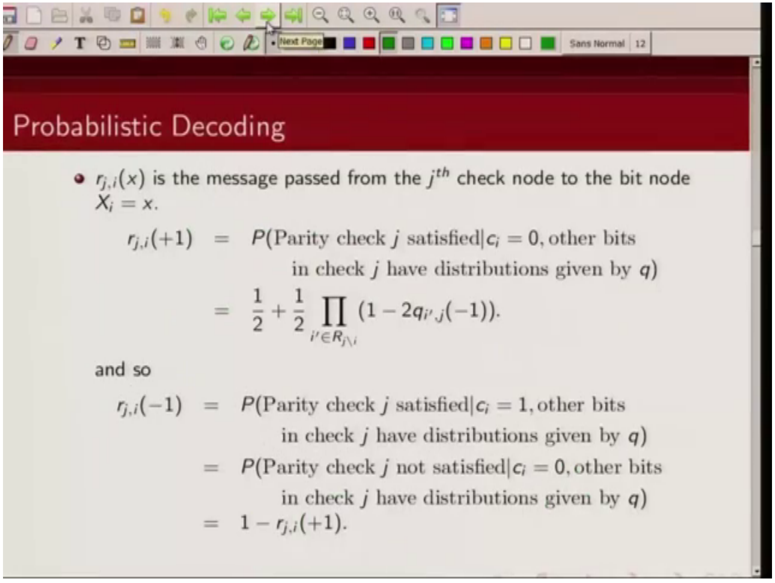
- Let's consider the term $P(S_i|c_i = 0, \mathbf{y})$. Given $c_i = 0$, S_i holds if each of w_c parity checks involving c_i has the property that the $w_r - 1$ bits in the check *other than* c_i have even parity.
- For parity check $j \in C_i$, the probability that the $\underline{w_r - 1 \text{ bits}}$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

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

$R_{j\setminus i}$

So

(Refer Slide Time 23:15)



Probabilistic Decoding

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

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

and so

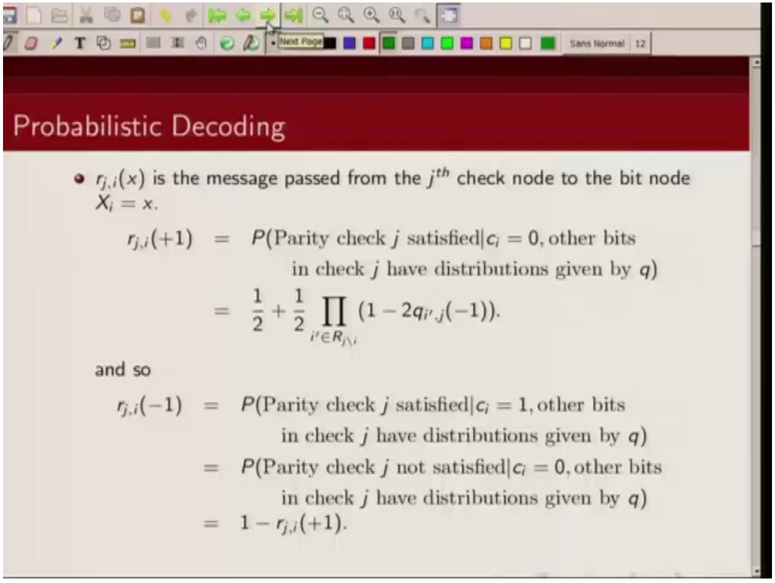
$$
\begin{aligned}
r_{j,i}(-1) &= P(\text{Parity check } j \text{ satisfied} | c_i = 1, \text{other bits} \\
&\qquad \text{in check } j \text{ have distributions given by } q) \\
&= P(\text{Parity check } j \text{ not satisfied} | c_i = 0, \text{other bits} \\
&\qquad \text{in check } j \text{ have distributions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

as we have said before we are writing, we are representing the L D P C code using standard graph and there are 2 types of information which are getting propagated. One is one sort of message is which is

(Refer Slide Time 23:29)

from the message nodes going to the variable, going to the parity check nodes. And there is one set of message from the, parity check nodes coming back to the message nodes. So

(Refer Slide Time 23:42)



Probabilistic Decoding

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

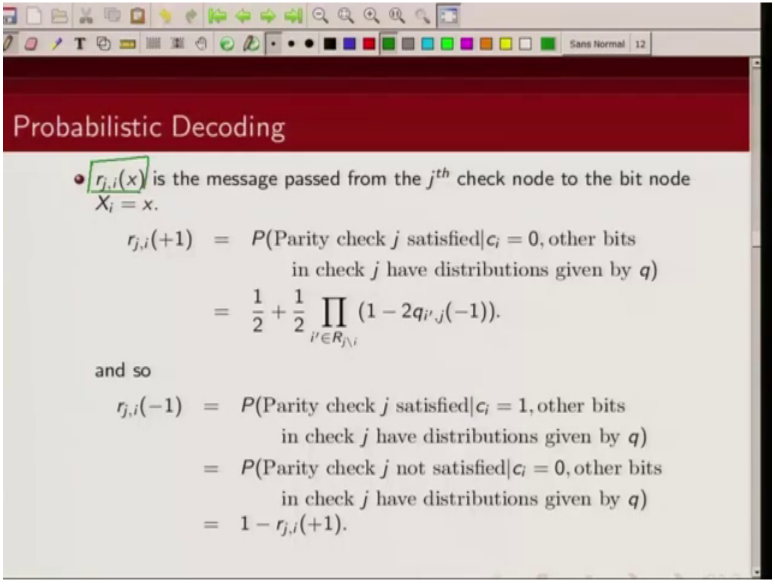
$$\begin{aligned} r_{j,i}(+1) &= P(\text{Parity check } j \text{ satisfied}|c_i = 0, \text{other bits} \\ &\quad \text{in check } j \text{ have distributions given by } q) \\ &= \frac{1}{2} + \frac{1}{2} \prod_{i' \in R_{j \setminus i}} (1 - 2q_{i',j}(-1)). \end{aligned}$$

and so

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

we are denoting by R j i, the message which is passed from jth parity check node to the ith bit. We are denoting this by R j i.

(Refer Slide Time 23:57)



Probabilistic Decoding

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

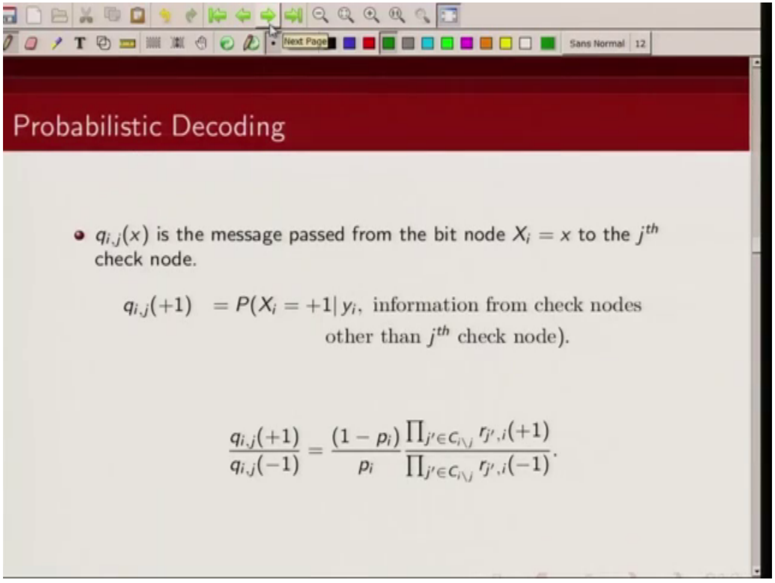
$$\begin{aligned} r_{j,i}(+1) &= P(\text{Parity check } j \text{ satisfied}|c_i = 0, \text{other bits} \\ &\quad \text{in check } j \text{ have distributions given by } q) \\ &= \frac{1}{2} + \frac{1}{2} \prod_{i' \in R_{j \setminus i}} (1 - 2q_{i',j}(-1)). \end{aligned}$$

and so

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

So what is R j i? R j i is the probability that j th check node is satisfied given x is plus 1. So this is a probability that j th check node is satisfied given c i is zero and other bits are given by distribution, given by this q. Now what is this q distribution? We will

(Refer Slide Time 24:24)



Probabilistic Decoding

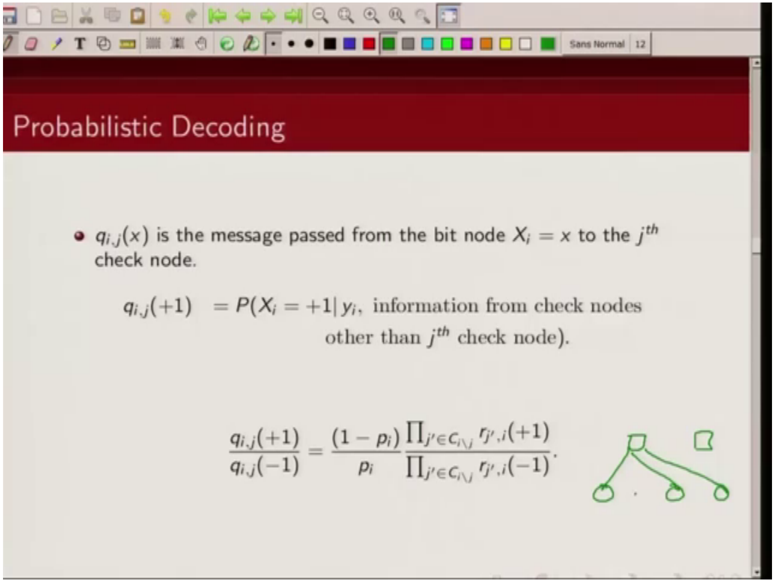
- $q_{i,j}(x)$ is the message passed from the bit node $X_i = x$ to the j^{th} check node.

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

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

come to, so there are 2 type of messages. As I said there is one message, so if I draw any Tanner graph let's say let's draw any Tanner graph, let's say this some graph I am

(Refer Slide Time 24:37)



Probabilistic Decoding

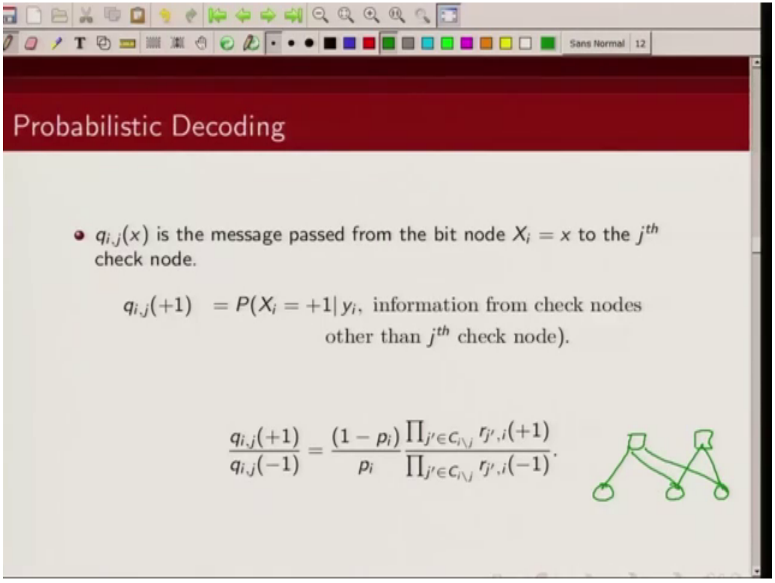
- $q_{i,j}(x)$ is the message passed from the bit node $X_i = x$ to the j^{th} check node.

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

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

drawing.

(Refer Slide Time 24:38)



Probabilistic Decoding

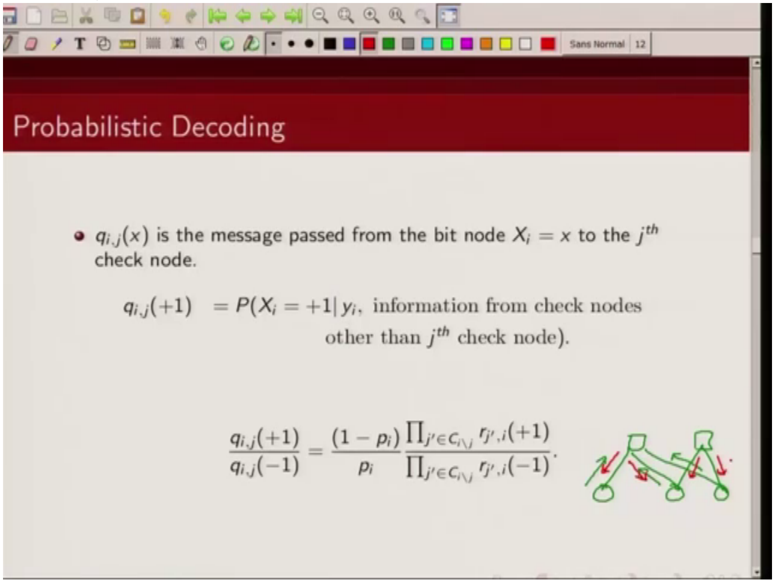
- $q_{i,j}(x)$ is the message passed from the bit node $X_i = x$ to the j^{th} check node.

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

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

So there is one set of messages which is going from message to this check nodes. Ok this is one sort of messages which are going like this. And there are another set of messages which is coming from the check node to the message bits. So

(Refer Slide Time 24:55)



Probabilistic Decoding

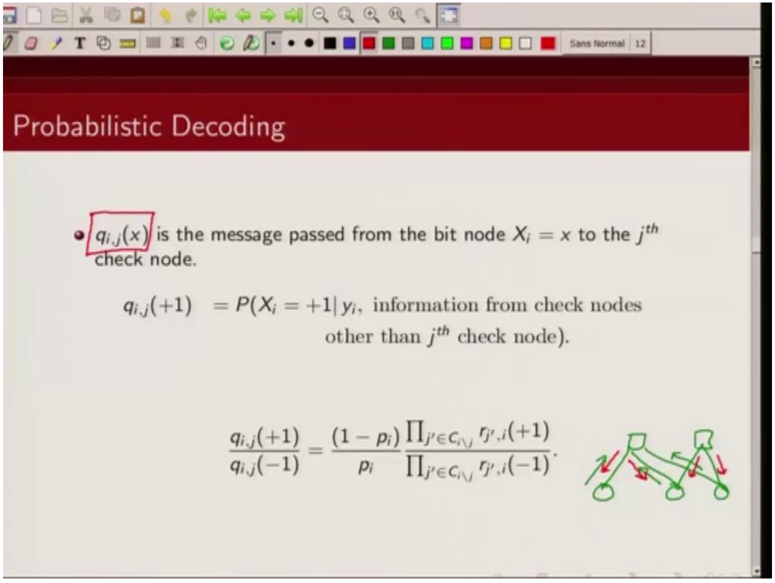
- $q_{i,j}(x)$ is the message passed from the bit node $X_i = x$ to the j^{th} check node.

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

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

we are denoting by q i’s

(Refer Slide Time 24:59)



Probabilistic Decoding

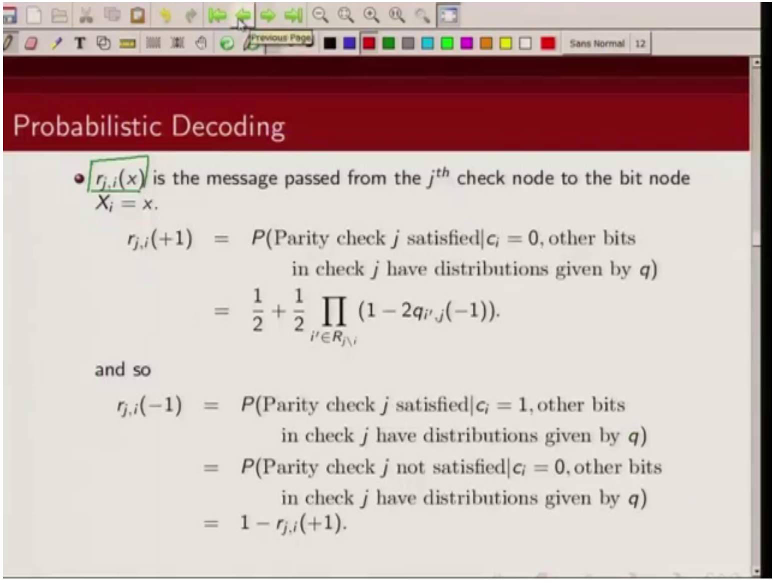
- $\boxed{q_{i,j}(x)}$ is the message passed from the bit node $X_i = x$ to the j^{th} check node.

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

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

the message which is passed from bit node to the jth check node and we are denoting

(Refer Slide Time 25:05)



Probabilistic Decoding

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

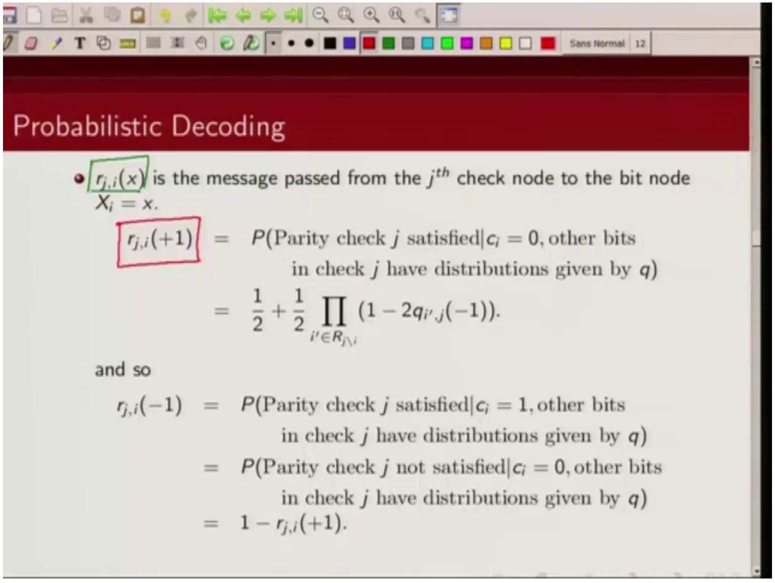
$$\begin{aligned} r_{j,i}(+1) &= P(\text{Parity check } j \text{ satisfied} | c_i = 0, \text{other bits} \\ &\qquad \text{in check } j \text{ have distributions given by } q) \\ &= \frac{1}{2} + \frac{1}{2} \prod_{i' \in R_{j \setminus i}} (1 - 2q_{i',j}(-1)). \end{aligned}$$

and so

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

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 25:18)



Probabilistic Decoding

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

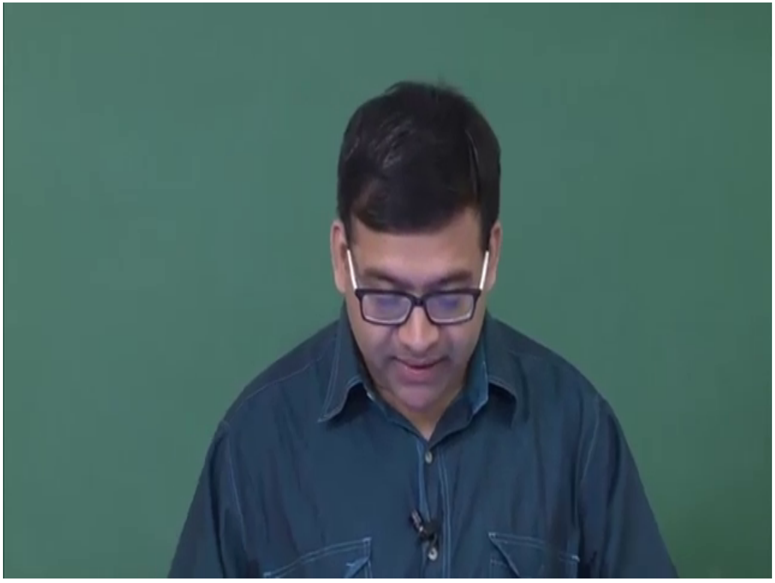
$$\boxed{r_{j,i}(+1)} = P(\text{Parity check } j \text{ satisfied} | c_i = 0, \text{other bits in check } j \text{ have distributions given by } q) = \frac{1}{2} + \frac{1}{2} \prod_{i' \in R_{j\setminus i}} (1 - 2q_{i',j}(-1)).$$

and so

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

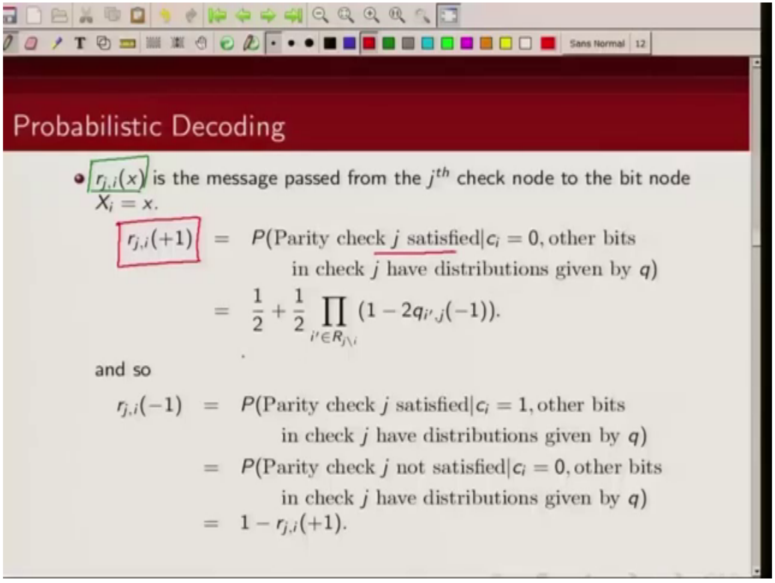
x being plus 1 which basically corresponds to c i, code bit being zero, this probability is defined as what is the probability that jth parity check constraint is satisfied given that c i, the ith bit is zero and other bits are given by, distribution given by q i. Now what is the probability that jth parity check constraint 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 constraint, they should have even parity

(Refer Slide Time 26:00)



and that

(Refer Slide Time 26:01)



Probabilistic Decoding

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

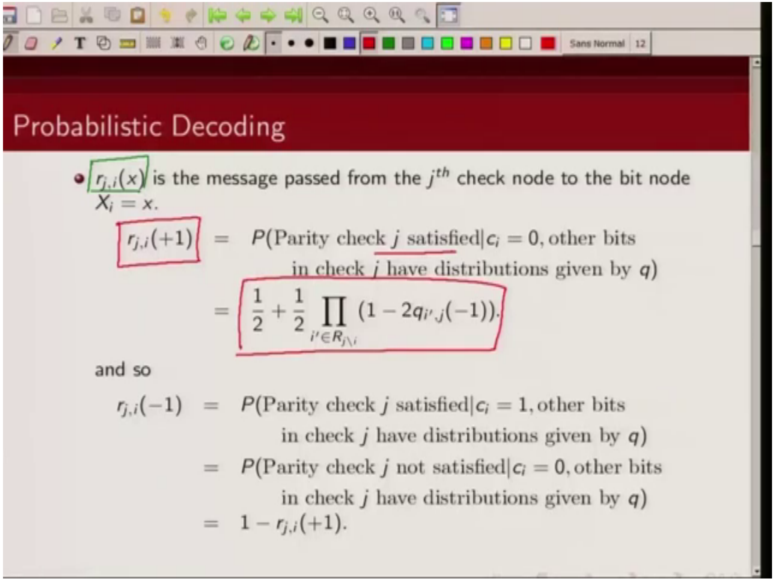
$$\boxed{r_{j,i}(+1)} = P(\text{Parity check } \underline{j \text{ satisfied}} | c_i = 0, \text{other bits in check } j \text{ have distributions given by } q) = \frac{1}{2} + \frac{1}{2} \prod_{i' \in R_{j \setminus i}} (1 - 2q_{i',j}(-1)).$$

and so

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

is given by this expression.

(Refer Slide Time 26:04)



Probabilistic Decoding

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

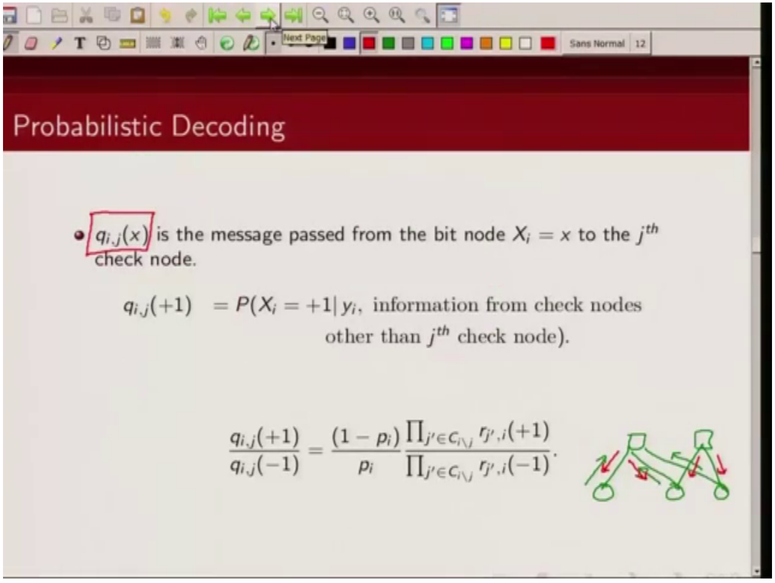
$$\boxed{r_{j,i}(+1)} = P(\text{Parity check } \underline{j \text{ satisfied}} | c_i = 0, \text{other bits in check } j \text{ have distributions given by } q) = \boxed{\frac{1}{2} + \frac{1}{2} \prod_{i' \in R_{j \setminus i}} (1 - 2q_{i',j}(-1)).}$$

and so

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

Similarly we can find out what's the probability that R j i is minus 1, this is 1 minus this probability. Now what

(Refer Slide Time 26:16)



Probabilistic Decoding

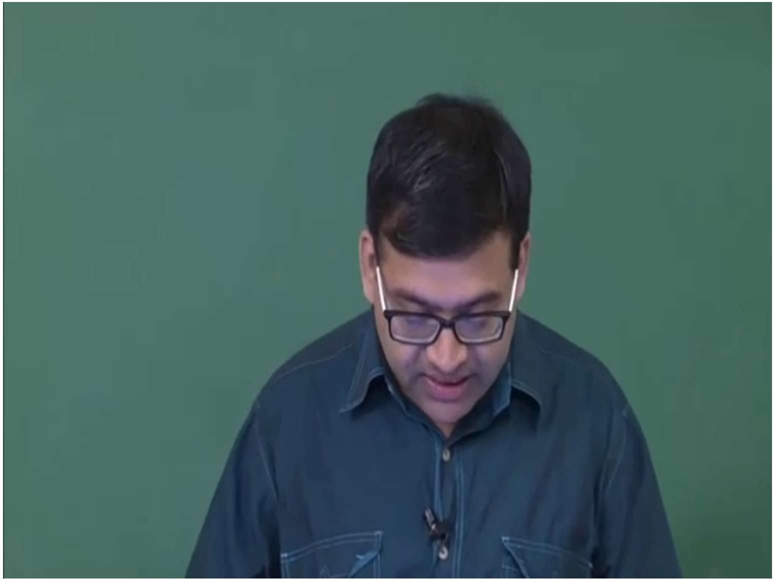
- $\boxed{q_{i,j}(x)}$ is the message passed from the bit node $X_i = x$ to the j^{th} check node.

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

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

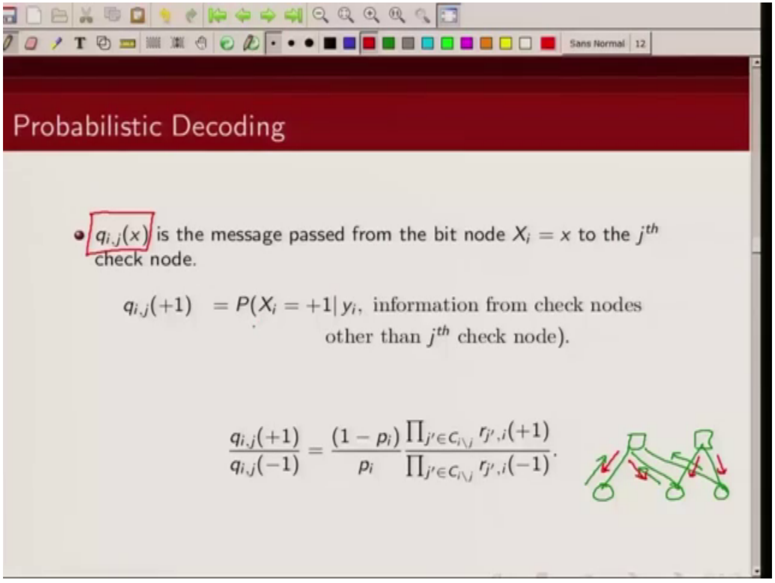
is this q i j? It's a message passed from the bit location i to the jth

(Refer Slide Time 26:24)



parity check node. And this is,

(Refer Slide Time 26:26)



Probabilistic Decoding

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

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

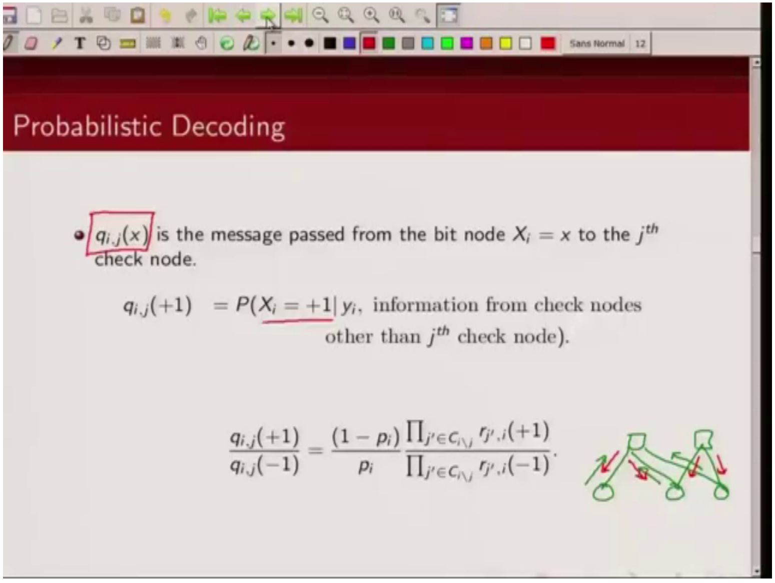
$$\frac{q_{i,j}(+1)}{q_{i,j}(-1)} = \frac{(1-p_i)}{p_i} \frac{\prod_{j' \in C_{i \setminus j}} r_{j',i}(+1)}{\prod_{j' \in C_{i \setminus j}} r_{j',i}(-1)}.$$

so q i j being plus 1 is given by what's the probability that x i is plus 1 given received sequence y i and information from parity check nodes other than the jth parity check node. So what's happening is

(Refer Slide Time 26:44)

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 27:05)



Probabilistic Decoding

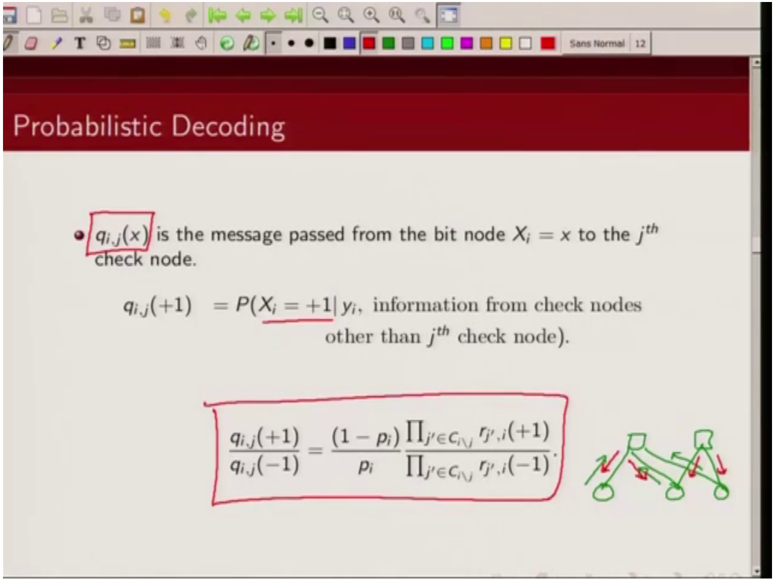
- $q_{i,j}(x)$ is the message passed from the bit node $X_i = x$ to the j^{th} check node.

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

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

And this probability is given by this expression.

(Refer Slide Time 27:12)



Probabilistic Decoding

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

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

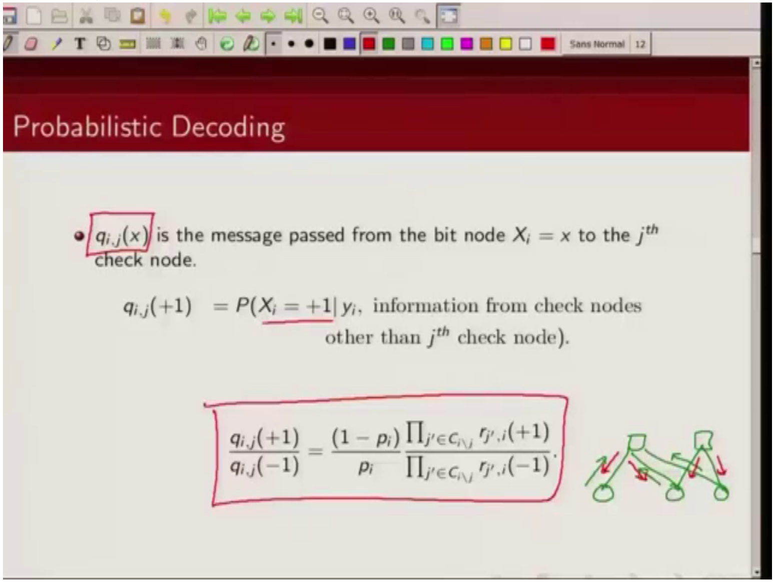
$$\boxed{\frac{q_{i,j}(+1)}{q_{i,j}(-1)} = \frac{(1-p_i)}{p_i}\frac{\prod_{j'\in C_{i\setminus j}} r_{j',i}(+1)}{\prod_{j'\in C_{i\setminus j}} r_{j',i}(-1)}.}$$

So there are 2 types of messages again, I repeat which are being propagated in this graph. One is

(Refer Slide Time 27:19)

the message from the message nodes to the check nodes and then check nodes are sending some information saying Ok whether these parity check constraints are satisfied or not given input bit is 1 or 0 and they are passing that information. So these information q is are passed from message nodes to the check nodes and these messages r is are being passed from the check nodes to the

(Refer Slide Time 27:46)



Probabilistic Decoding

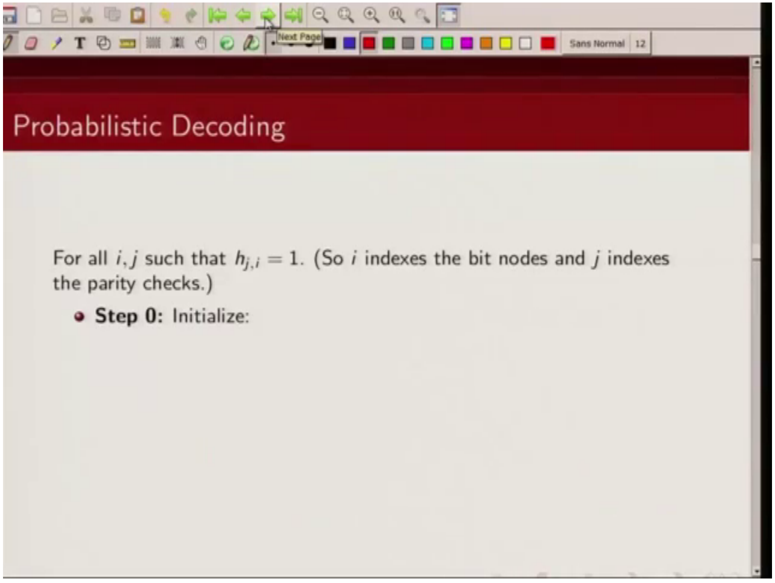
- $\boxed{q_{i,j}(x)}$ is the message passed from the bit node $X_i = x$ to the j^{th} check node.

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

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

message nodes.

(Refer Slide Time 27:47)



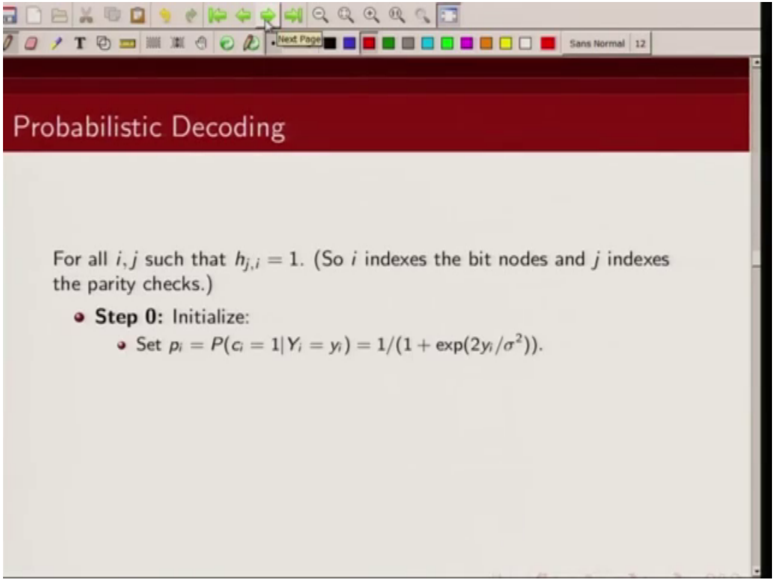
Probabilistic Decoding

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 27:52)



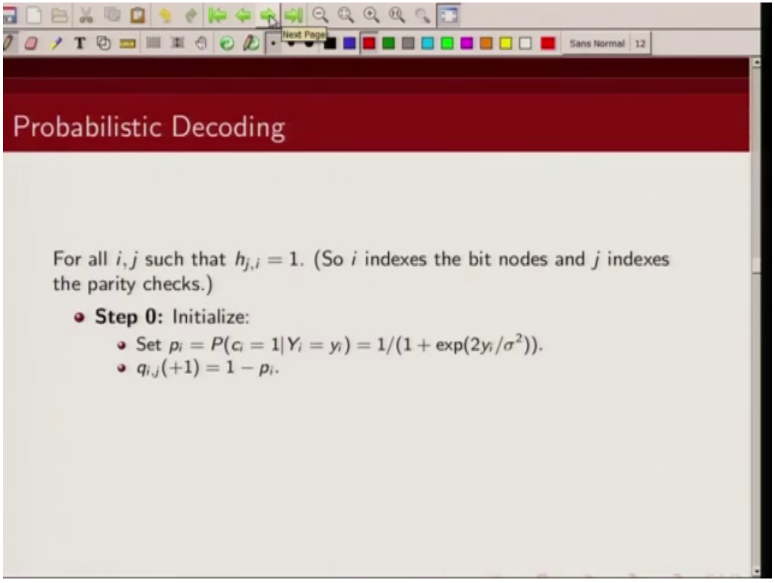
Probabilistic Decoding

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 28:00)



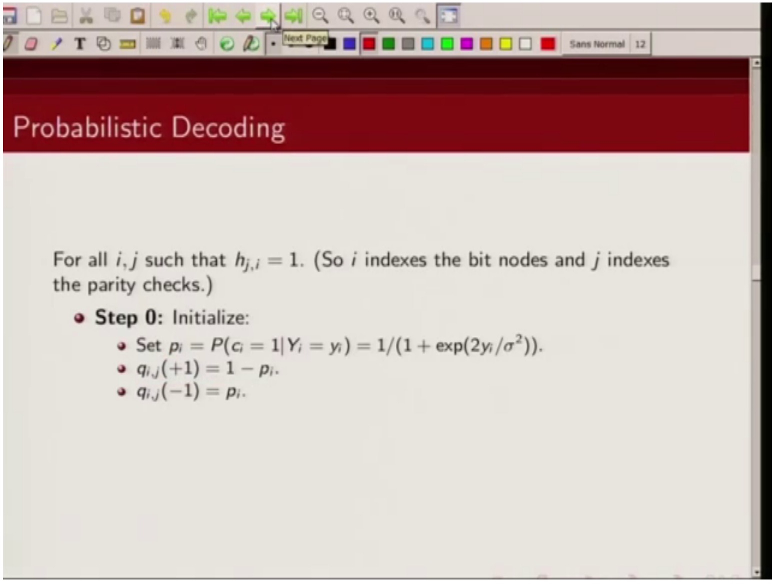
Probabilistic Decoding

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

So we calculate this p i and we calculate this q is. These are messages we will send from the bit nodes to the party check nodes,

(Refer Slide Time 28:12)



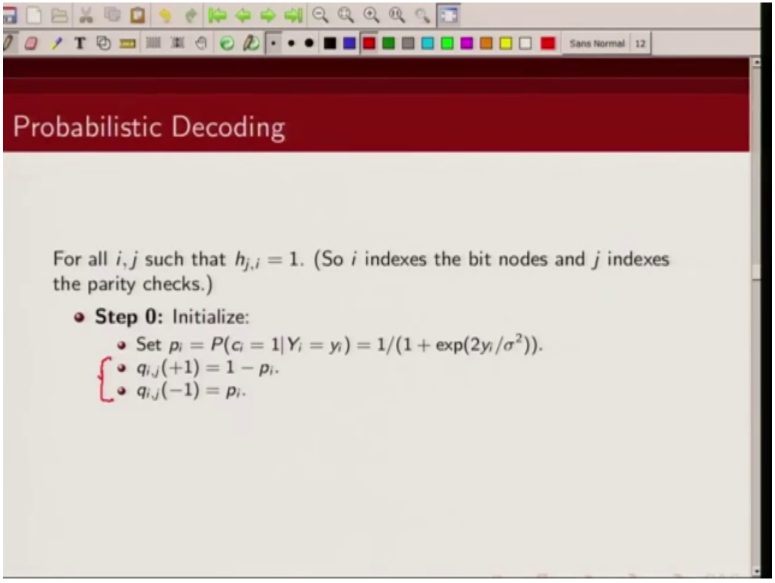
Probabilistic Decoding

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

Ok. So that's the first step, initialization step that we are calculating. The initial messages that the bit nodes will send to the

(Refer Slide Time 28:23)



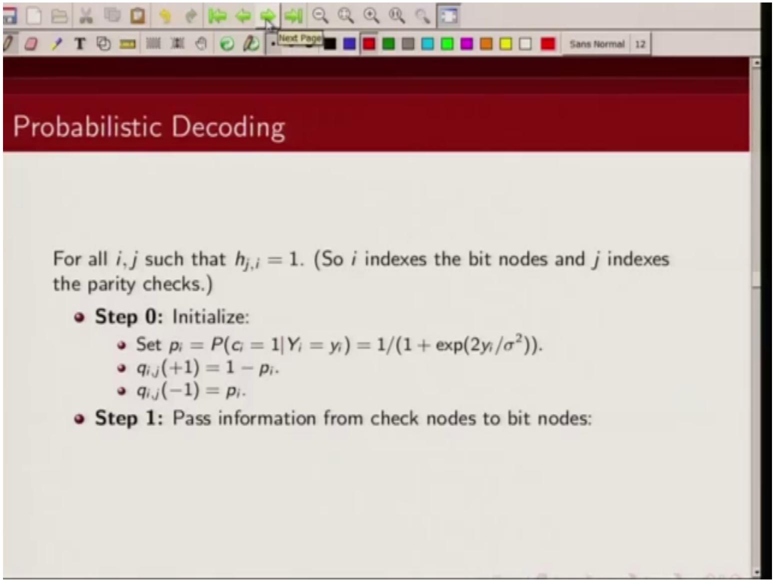
Probabilistic Decoding

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

check nodes and that is basically based on the channel likelihood values. It is given by this expression.

(Refer Slide Time 28:35)



Probabilistic Decoding

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

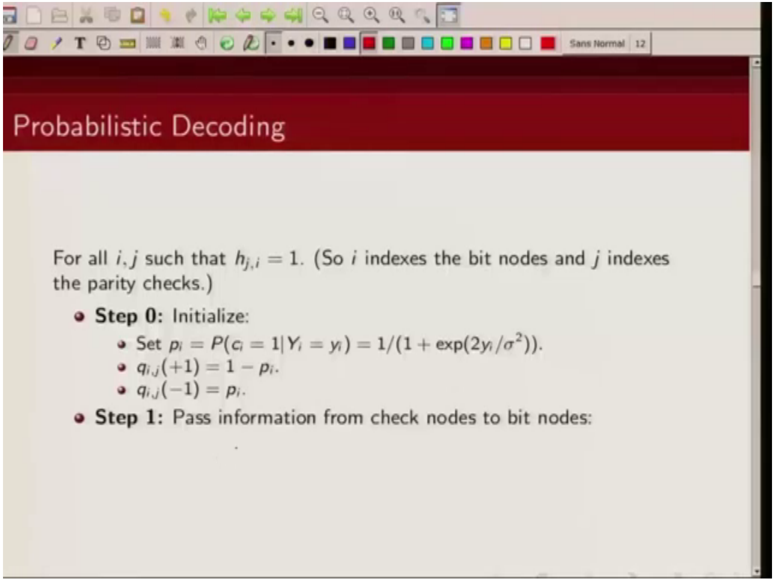
- **Step 0:** Initialize:
  - Set $p_i = P(c_i = 1|Y_i = y_i) = 1/(1 + \exp(2y_i/\sigma^2))$.
  - $q_{i,j}(+1) = 1 - p_i$.
  - $q_{i,j}(-1) = p_i$.
- **Step 1:** Pass information from check nodes to bit nodes:

Next is, once these messages are assigned to check nodes, then check nodes will do local computation and it will

(Refer Slide Time 28:42)

r is back to the bit nodes.

(Refer Slide Time 28:46)



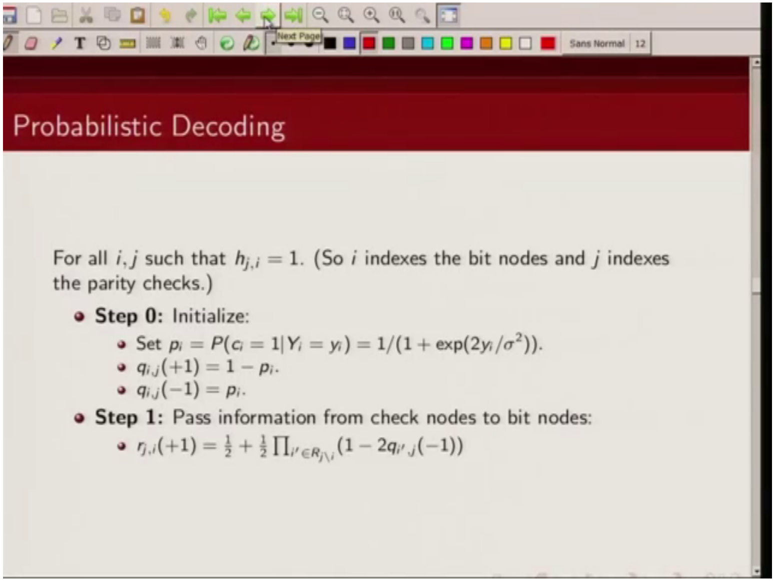
Probabilistic Decoding

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 passing information from check nodes to bit nodes, so check nodes

(Refer Slide Time 28:49)



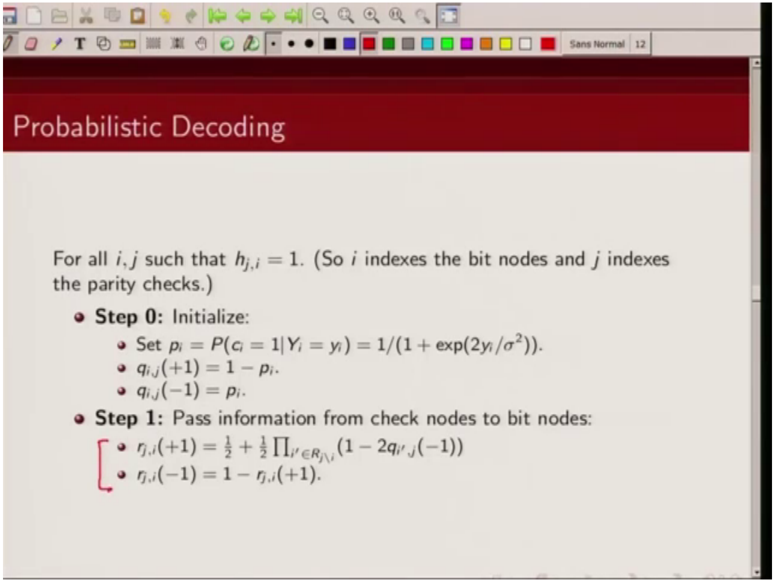
Probabilistic Decoding

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

are going to pass this information r is back to the bit nodes and this is given by this expression.

(Refer Slide Time 28:57)



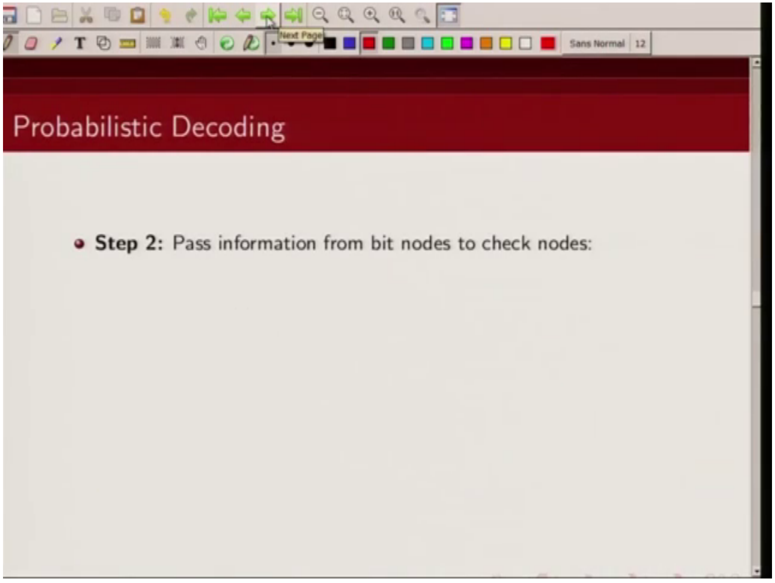
Probabilistic Decoding

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

Now

(Refer Slide Time 29:00)



Probabilistic Decoding

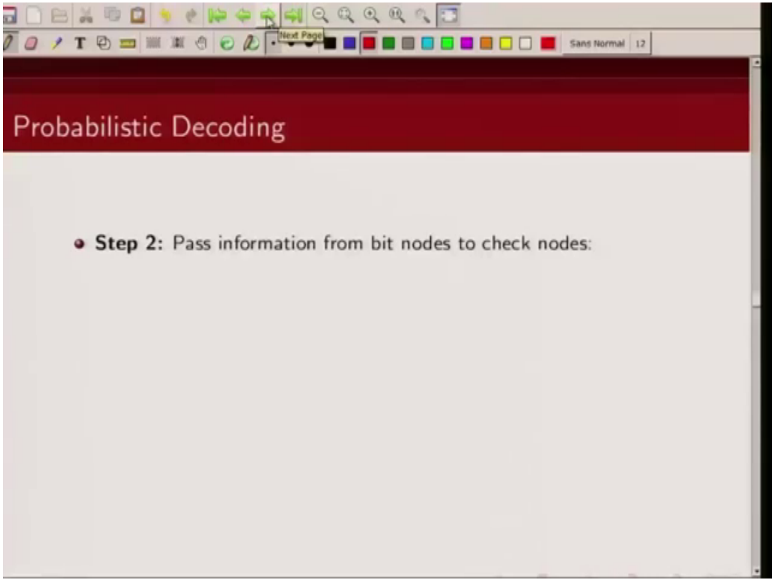
- **Step 2:** Pass information from bit nodes to check nodes:

once you get this updated r is

(Refer Slide Time 29:02)

from various check nodes then this q is are updated.

(Refer Slide Time 29:07)

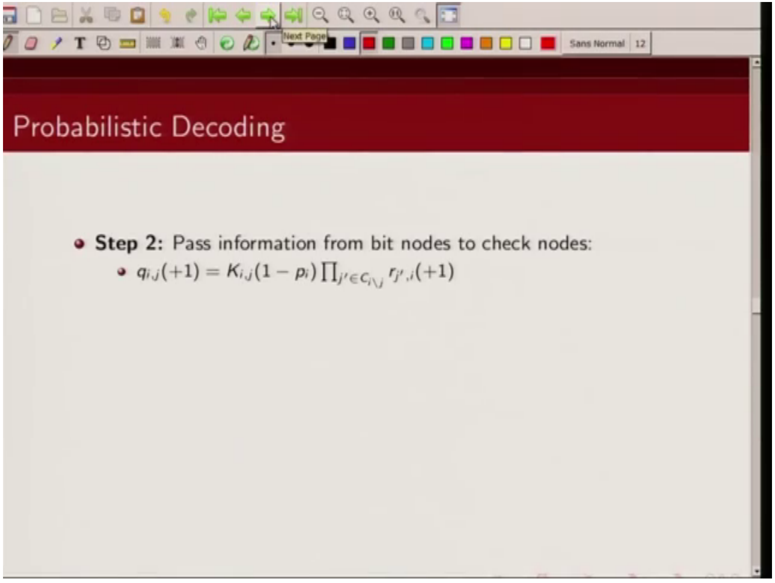


Probabilistic Decoding

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

So you are going to then

(Refer Slide Time 29:10)

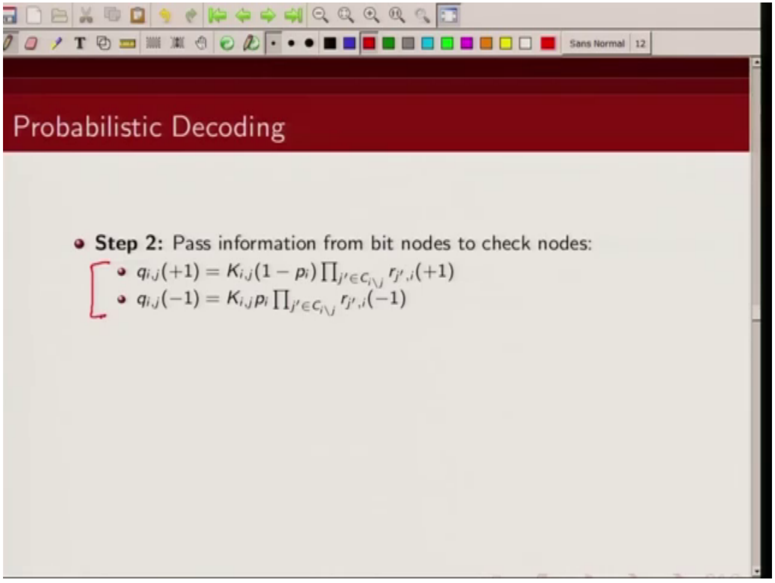


Probabilistic Decoding

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

send modified information, this q is to the parity check nodes and this is basically given

(Refer Slide Time 29:18)

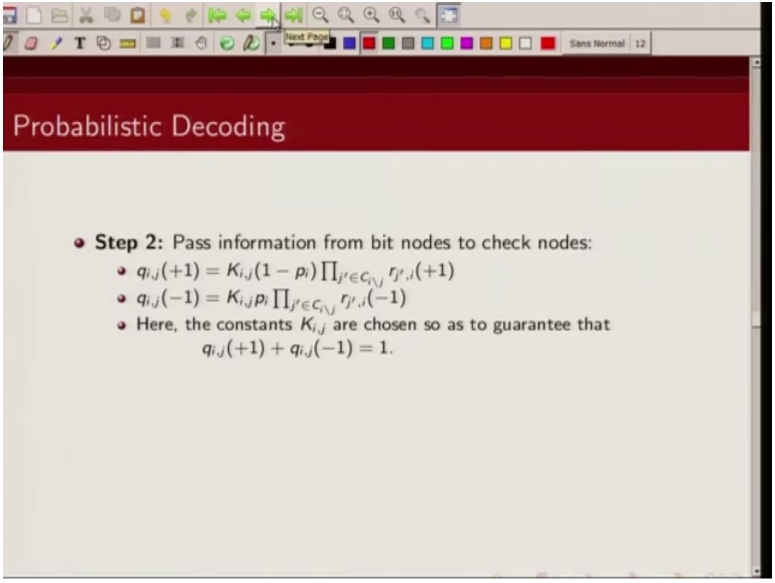


Probabilistic Decoding

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

by this. You can do some normalization. These are k is are just

(Refer Slide Time 29:22)

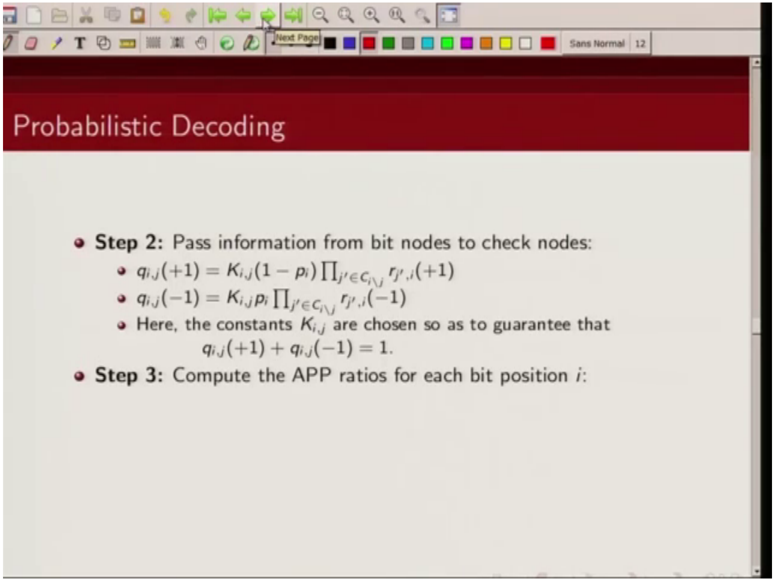


Probabilistic Decoding

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

some constant. You can do some normalization. And finally we will compute the

(Refer Slide Time 29:28)

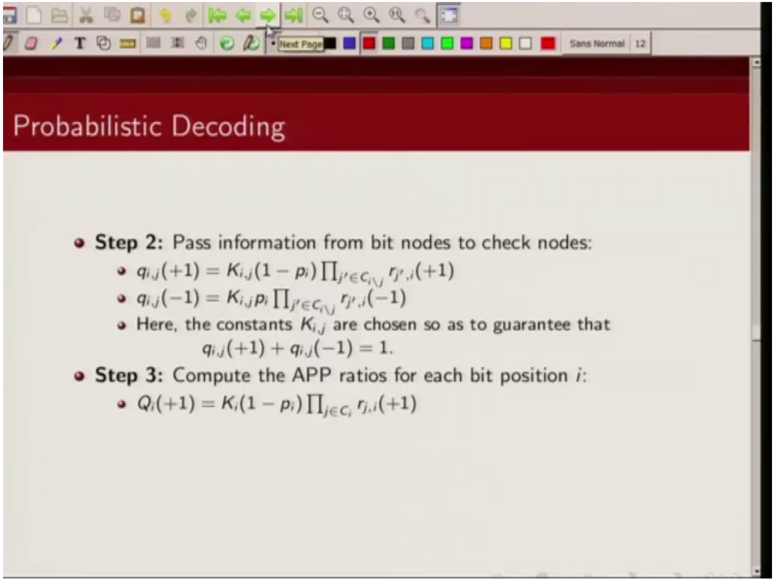


Probabilistic Decoding

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

a posteriori probability and that's

(Refer Slide Time 29:32)

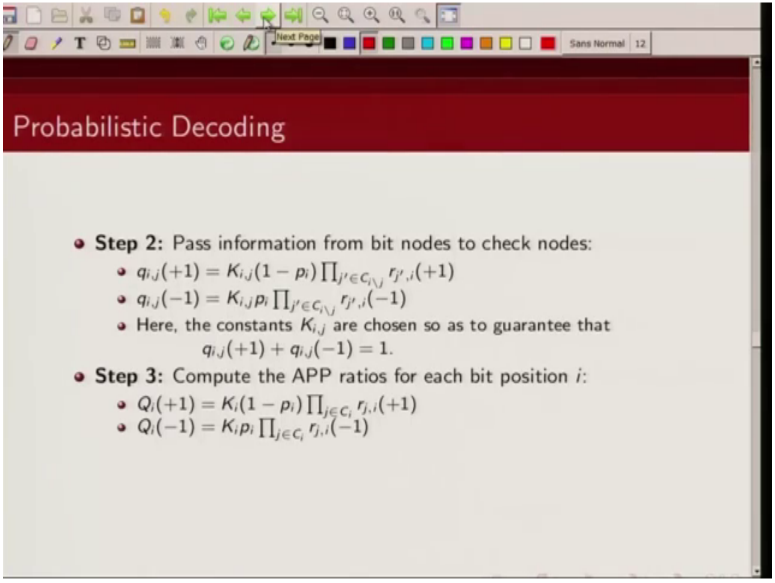


Probabilistic Decoding

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

given by

(Refer Slide Time 29:34)

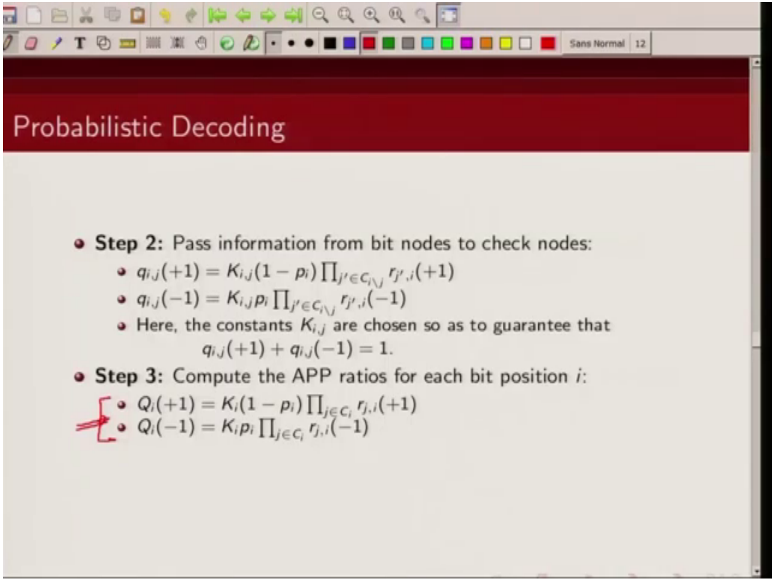


Probabilistic Decoding

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

this expression. And this we have derived earlier

(Refer Slide Time 29:39)

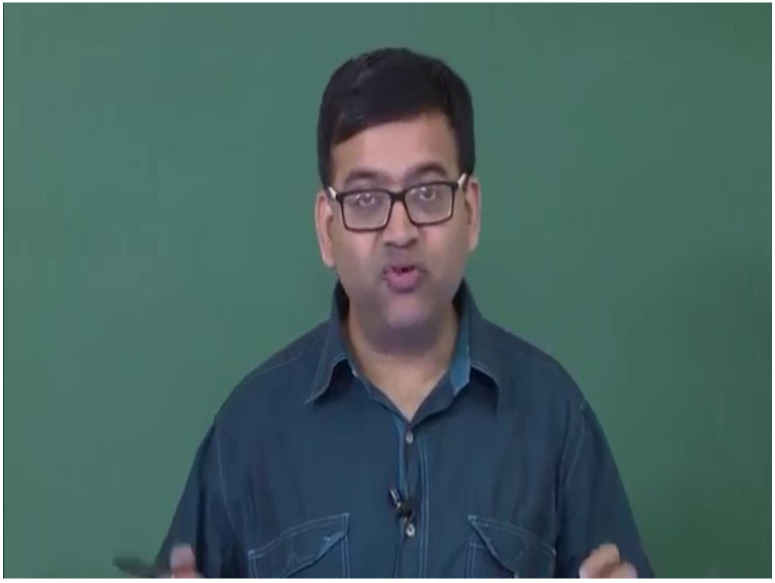


Probabilistic Decoding

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

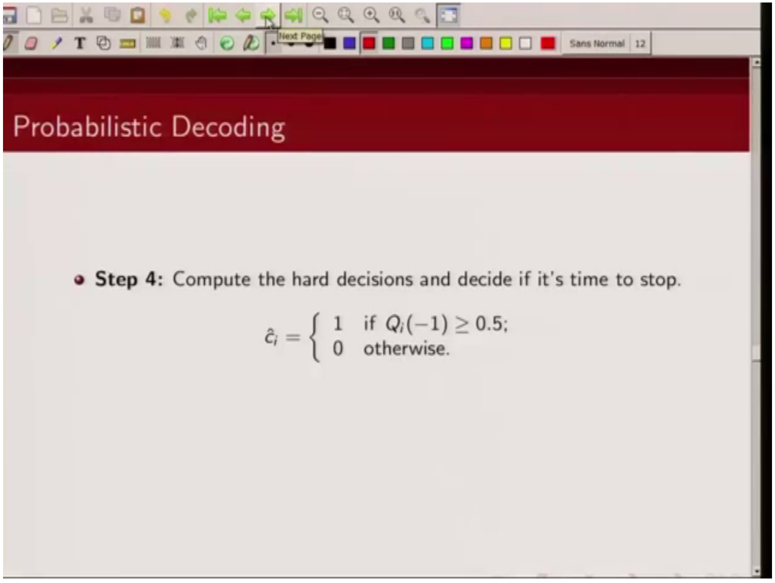
in the lecture, Ok.

(Refer Slide Time 29:42)



So to repeat basically how this whole process is going, you have this received values from the channel. Based on that you compute initial p is, and q is. Now these information are assigned to check nodes. Now check nodes do local computation, that what's the probability that check node will be satisfied given a particular bit c i is zero or 1. 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 account to update its q is. And this process is continued in an iterative fashion until all the parity check constraints are satisfied.

(Refer Slide Time 30:36)



Probabilistic Decoding

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

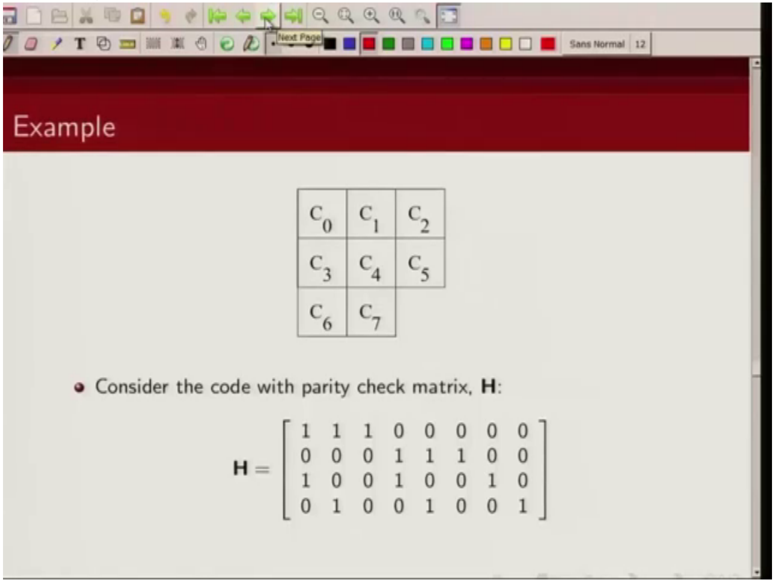
$$\hat{c}_i = \begin{cases} 1 & \text{if } Q_i(-1) \geq 0.5; \\ 0 & \text{otherwise.} \end{cases}$$

So finally basically once you compute the a posteriori probability then q i being minus 1 is more than 1, you decide in favor of 1 or you decide

(Refer Slide Time 30:45)

in favor of 0. So

(Refer Slide Time 30:47)



Example

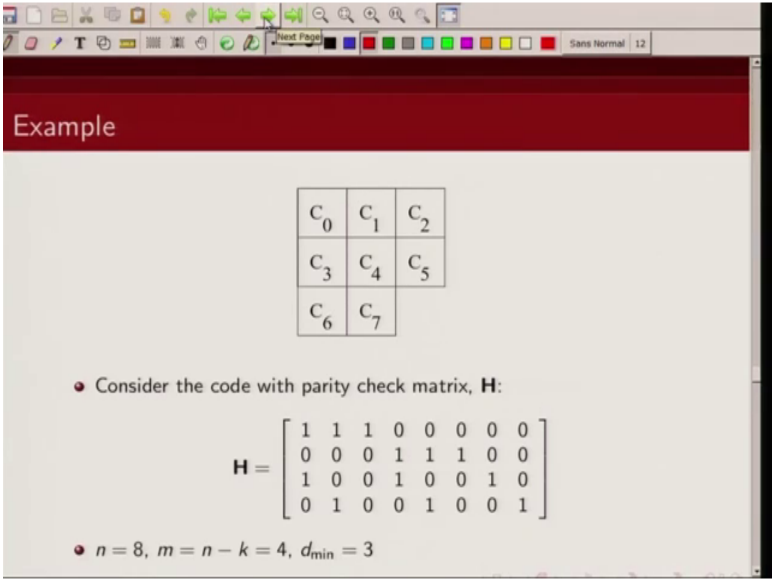
C_0	C_1	C_2
C_3	C_4	C_5
C_6	C_7	

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

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

let us take an example to illustrate the decoding algorithm. So we have a low density parity check matrix given by this. We have 8 coded bits.

(Refer Slide Time 31:00)



Example

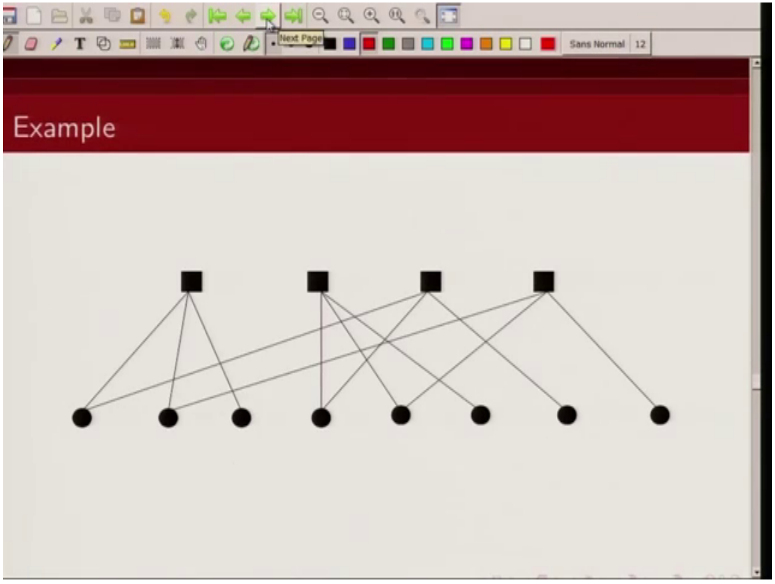
c_0	c_1	c_2
c_3	c_4	c_5
c_6	c_7	

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

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

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

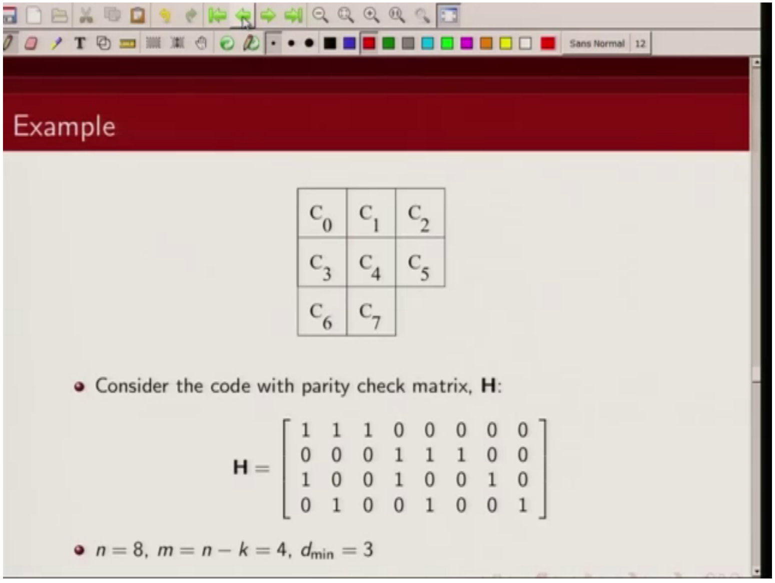
(Refer Slide Time 31:02)



Example

This is the Tanner graph corresponding to the parity check matrix. We can just quickly check it.

(Refer Slide Time 31:08)



Example

c_0	c_1	c_2
c_3	c_4	c_5
c_6	c_7	

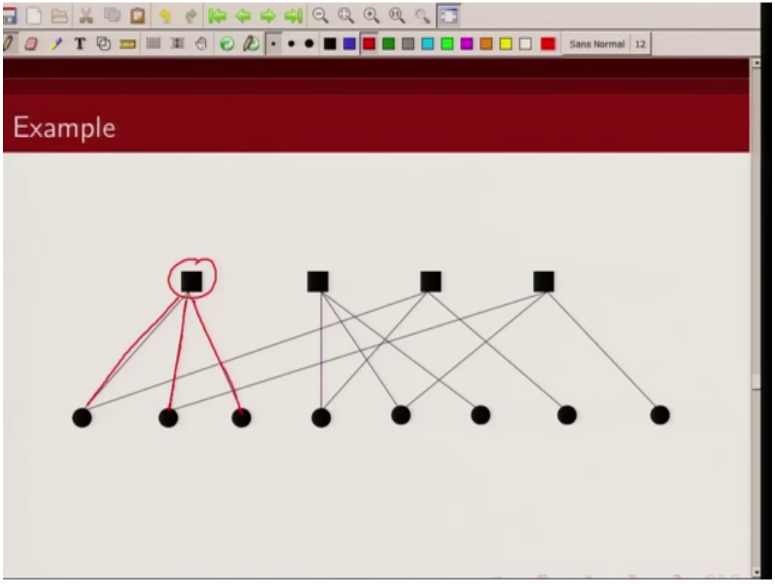
- Consider the code with parity check matrix, **H**:

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

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

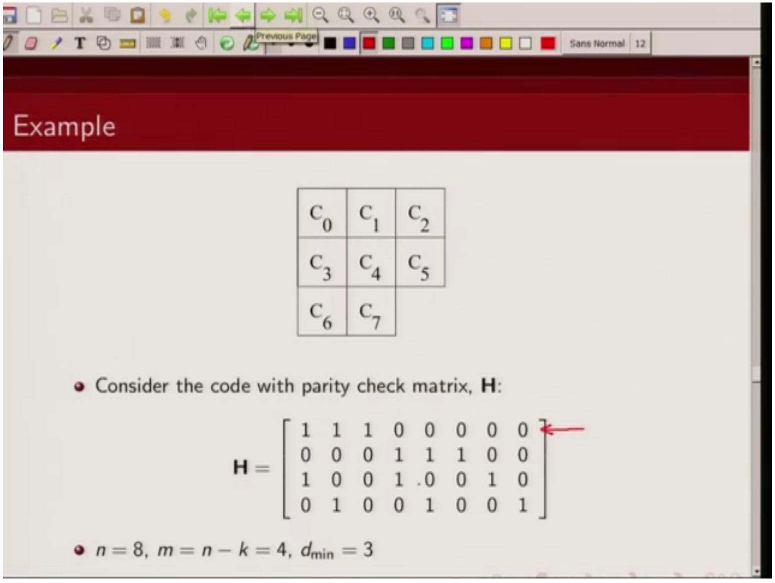
So the first parity check equation involves these 3 bits, first 3 bits. So you can see the first parity check equation involves this, this and this.

(Refer Slide Time 31:20)



Example

(Refer Slide Time 31:21)



Example

c_0	c_1	c_2
c_3	c_4	c_5
c_6	c_7	

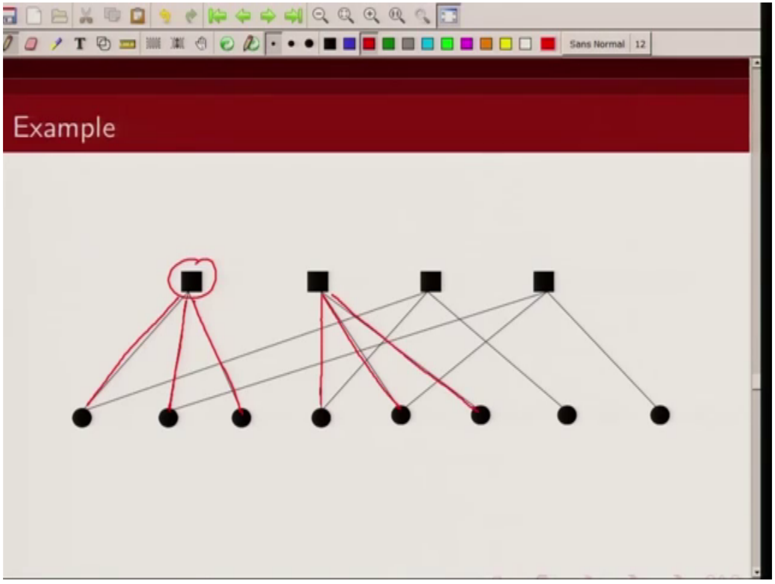
- Consider the code with parity check matrix, **H**:

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

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

Second parity check involves fourth, fifth and sixth bit. That’s second parity check involves fourth, fifth and sixth bit.

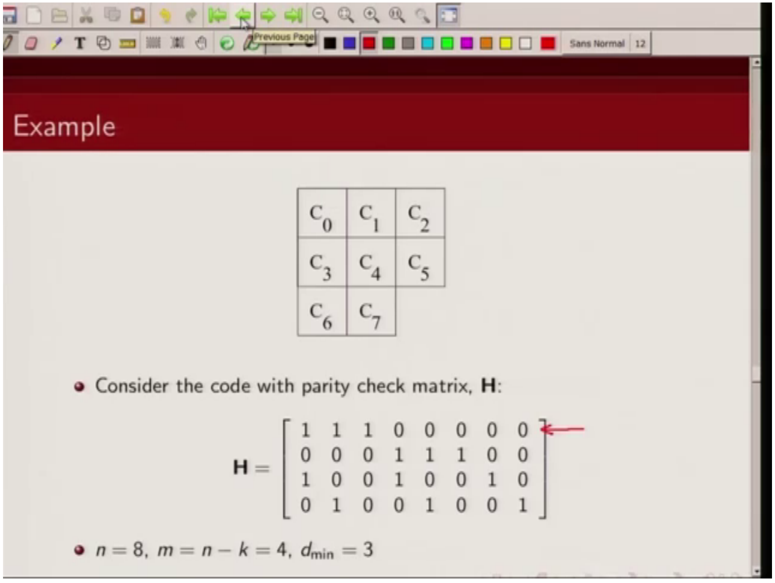
(Refer Slide Time 31:32)



Example

This involves

(Refer Slide Time 31:34)



Example

c_0	c_1	c_2
c_3	c_4	c_5
c_6	c_7	

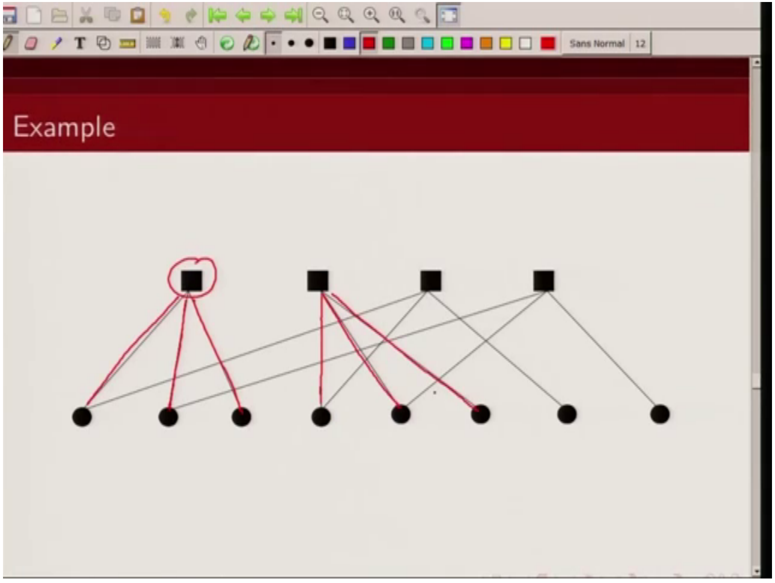
- Consider the code with parity check matrix, **H**:

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

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

first, fourth and seventh. And that's

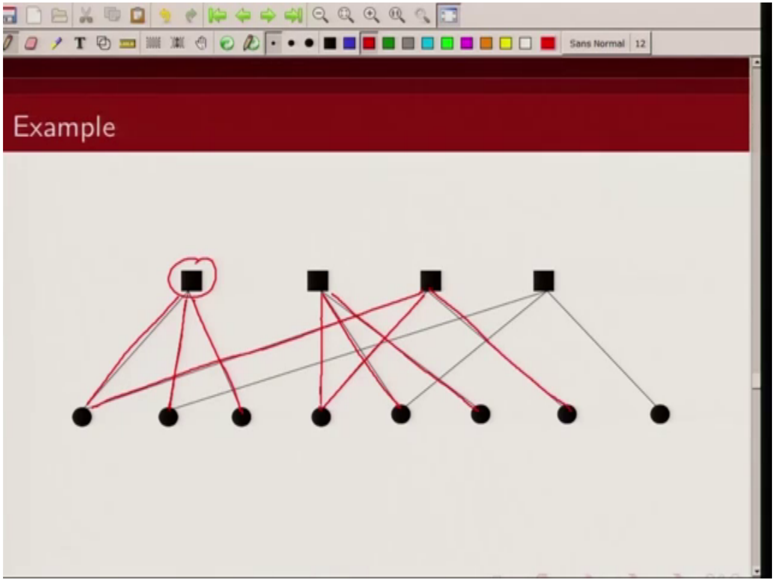
(Refer Slide Time 31:41)



Example

first, fourth and seventh

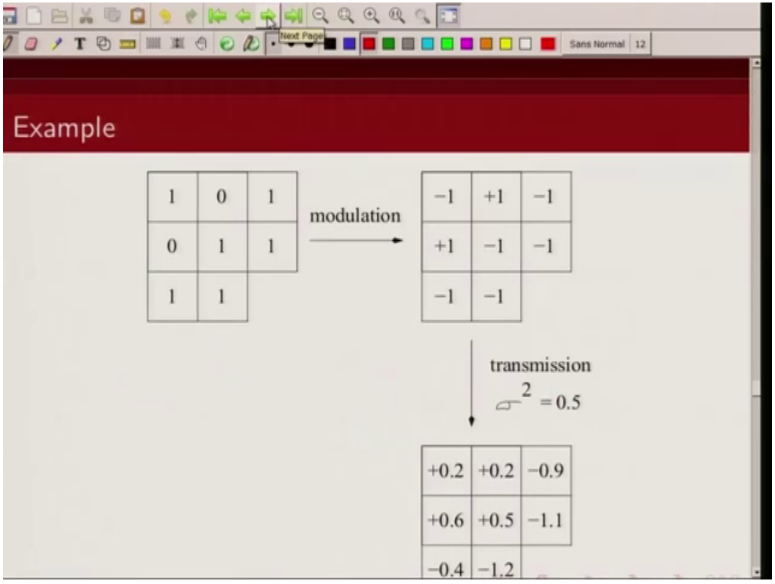
(Refer Slide Time 31:47)



Example

and similarly we can verify for this also, Ok. So this is Tanner graph corresponding to the parity check matrix that we have drawn.

(Refer Slide Time 31:57)



Example

1	0	1
0	1	1
1	1	

modulation

−1	+1	−1
+1	−1	−1
−1	−1	

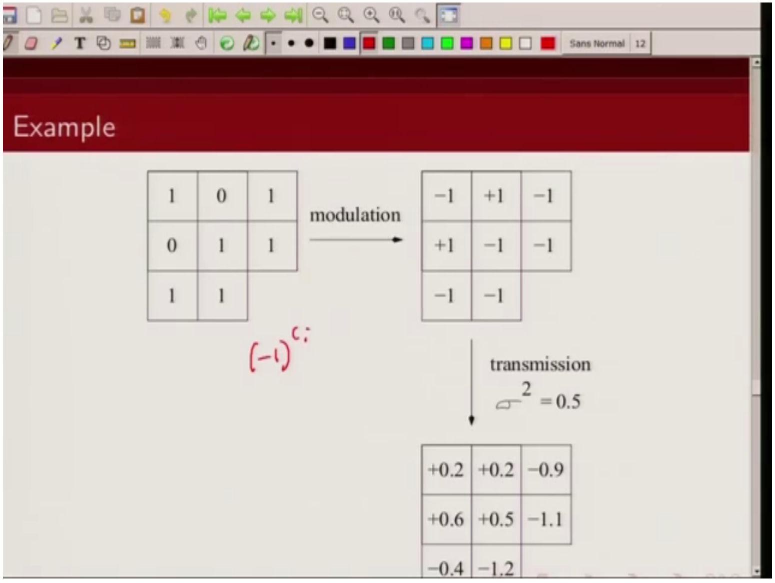
transmission

$\sigma^2 = 0.5$

+0.2	+0.2	−0.9
+0.6	+0.5	−1.1
−0.4	−1.2	

So these are the bits, 8 bits which were transmitted now after modulation, because we were mapping them as minus 1 c i

(Refer Slide Time 32:08)



Example

1	0	1
0	1	1
1	1	

modulation

−1	+1	−1
+1	−1	−1
−1	−1	

$(-1)^{c_i}$

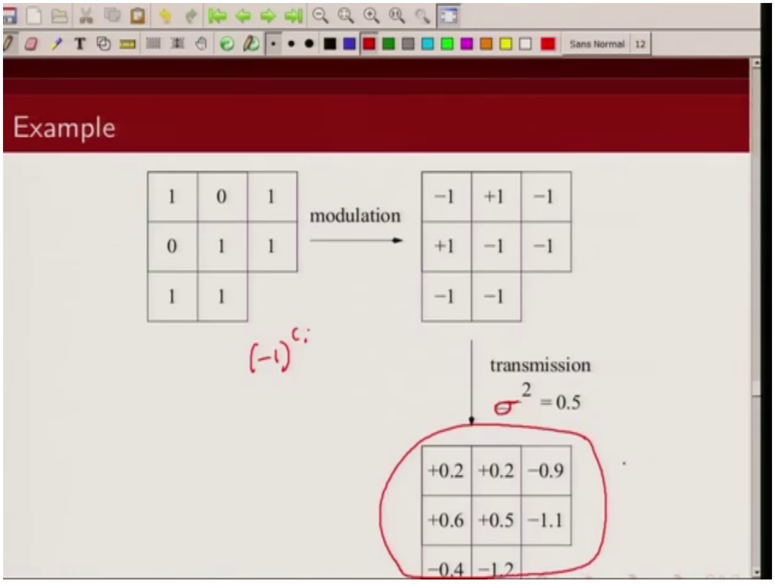
transmission

$\sigma^2 = 0.5$

+0.2	+0.2	−0.9
+0.6	+0.5	−1.1
−0.4	−1.2	

so 1 is getting mapped to minus 1 and 0 is getting mapped to plus 1. So these are the modulated bits and noise variance was point 5. So what we get is these are the received values, so these are my

(Refer Slide Time 32:25)



Example

1	0	1
0	1	1
1	1	

modulation

−1	+1	−1
+1	−1	−1
−1	−1	

$(-1)^{c_i}$

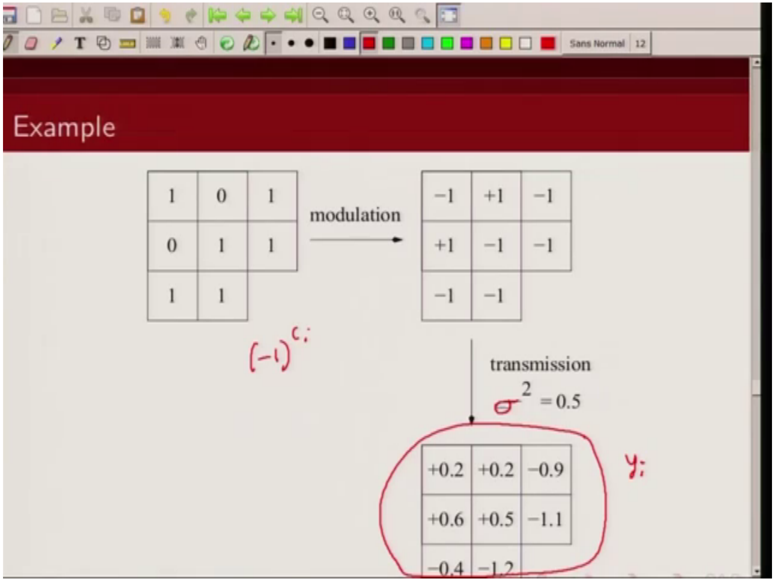
transmission

$\sigma^2 = 0.5$

+0.2	+0.2	−0.9
+0.6	+0.5	−1.1
−0.4	−1.2	

y is. Ok

(Refer Slide Time 32:28)



Example

1	0	1
0	1	1
1	1	

$\xrightarrow{\text{modulation}}$

−1	+1	−1
+1	−1	−1
−1	−1	

$(-1)^{c_i}$

transmission

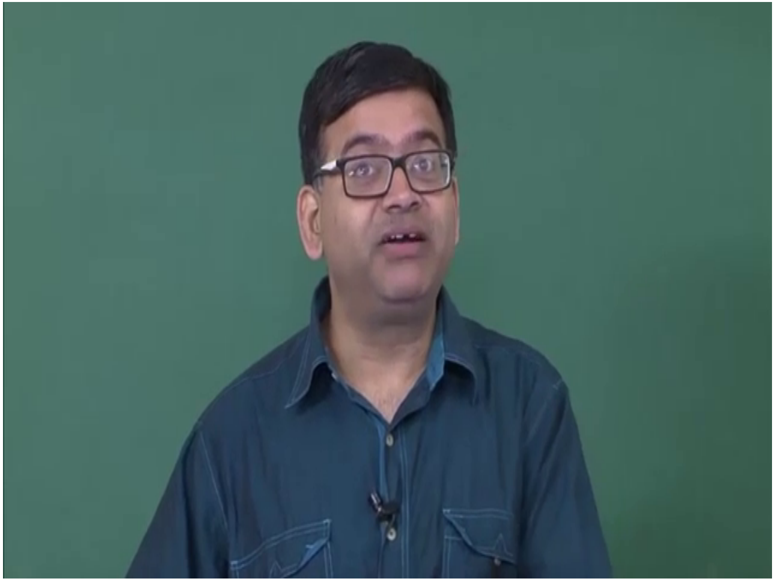
$\sigma^2 = 0.5$

+0.2	+0.2	−0.9
+0.6	+0.5	−1.1
−0.4	−1.2	

y_i

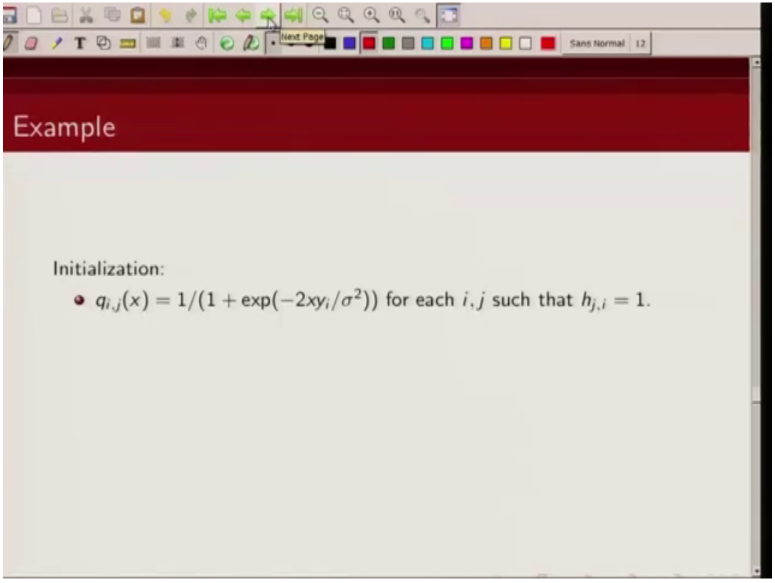
so what is the first step? I will take

(Refer Slide Time 32:32)



these y is and I will compute my p is and q is, initial p is.

(Refer Slide Time 32:36)



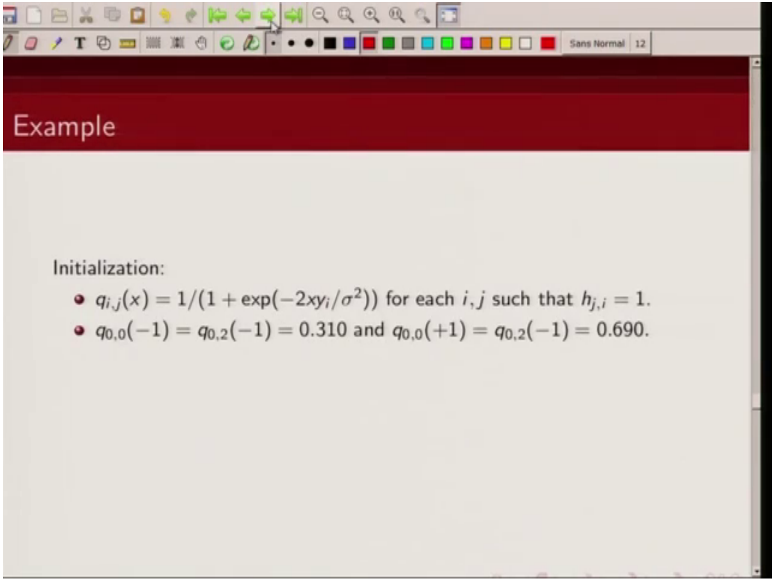
Example

Initialization:

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

So I am going to first compute my initial q is and that's basically given by this expression.

(Refer Slide Time 32:44)



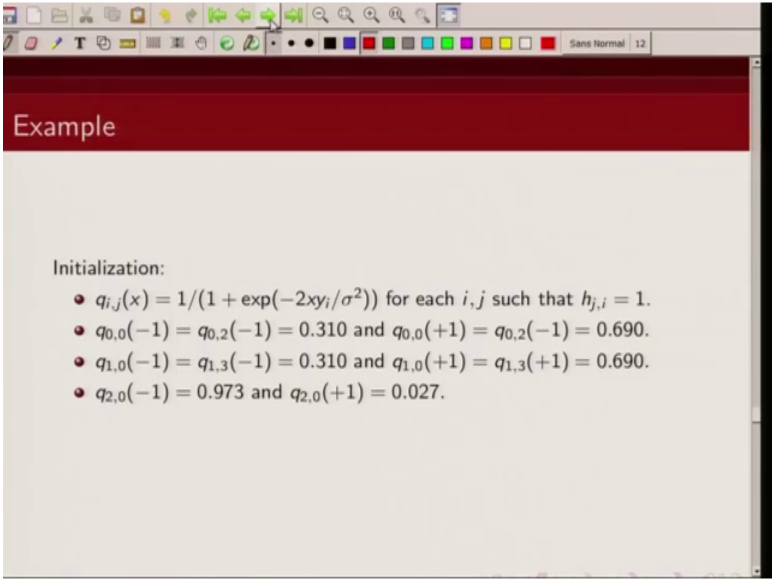
Example

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 $q_{0,0}(+1) = q_{0,2}(-1) = 0.690$.

So I am just stating

(Refer Slide Time 32:46)



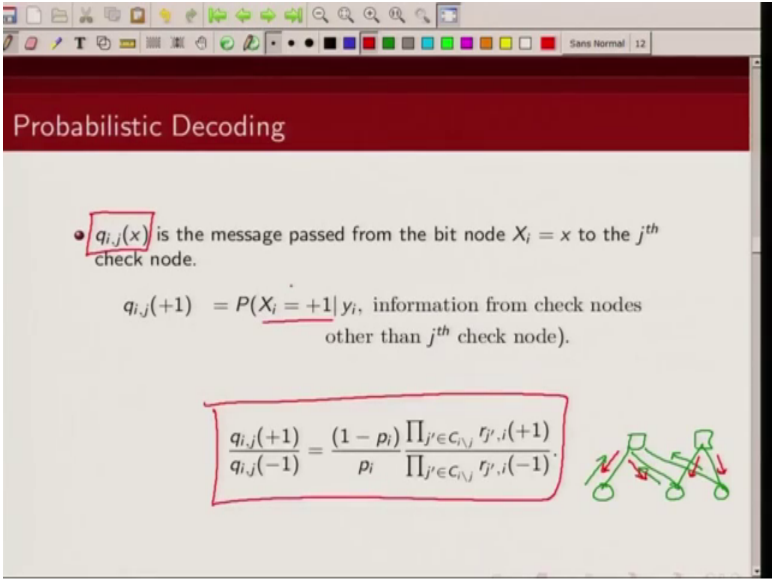
Example

Initialization:

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

this, so I will compute these q is. Now again, just pay little attention to what q is are. q i

(Refer Slide Time 32:59)



Probabilistic Decoding

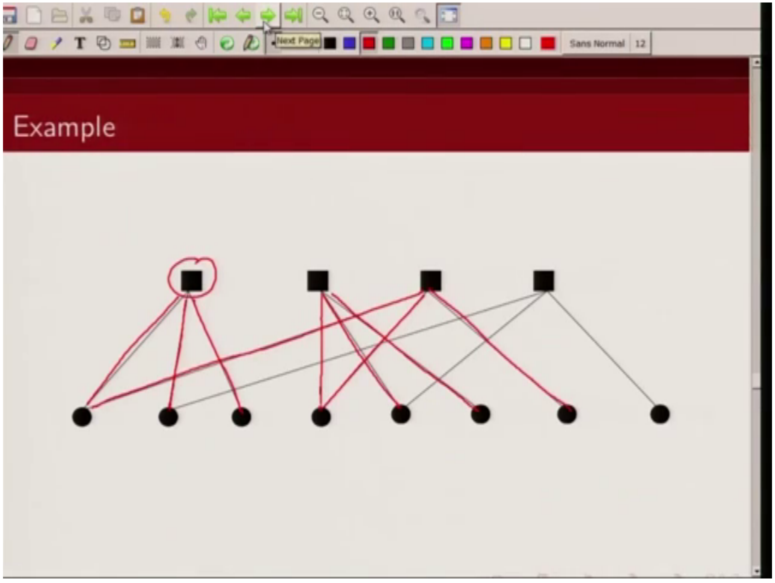
- $\boxed{q_{i,j}(x)}$ is the message passed from the bit node $X_i = x$ to the j^{th} check node.

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

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

is the message passed from ith bit to the jth parity check constraint. So if you label your nodes from 0, 1, 2, 3, 4, similarly label parity check constraints so what you are going to notice is, you need to compute your q is for in this example, so

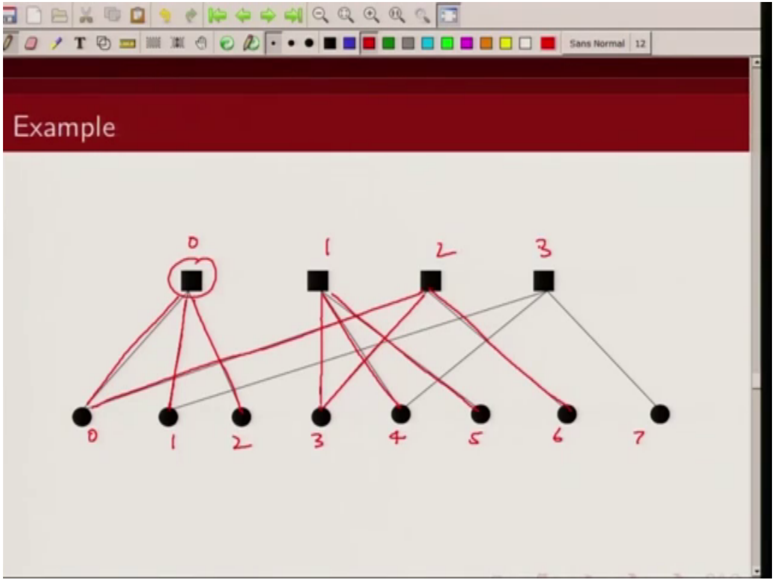
(Refer Slide Time 33:26)



Example

this is 0,1, 2,3, 4, 5,6, 7. Similarly it is 0, 1, 2, 3. So you will

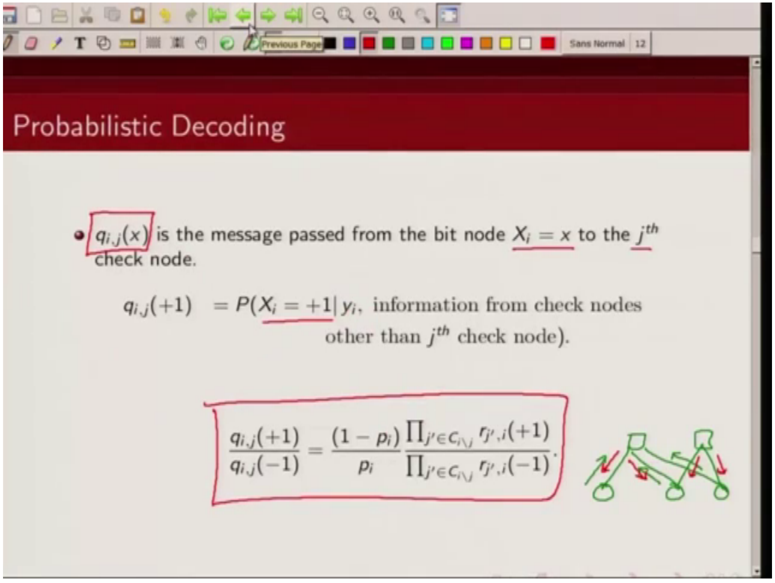
(Refer Slide Time 33:35)



Example

compute q 0 0,

(Refer Slide Time 33:40)



Probabilistic Decoding

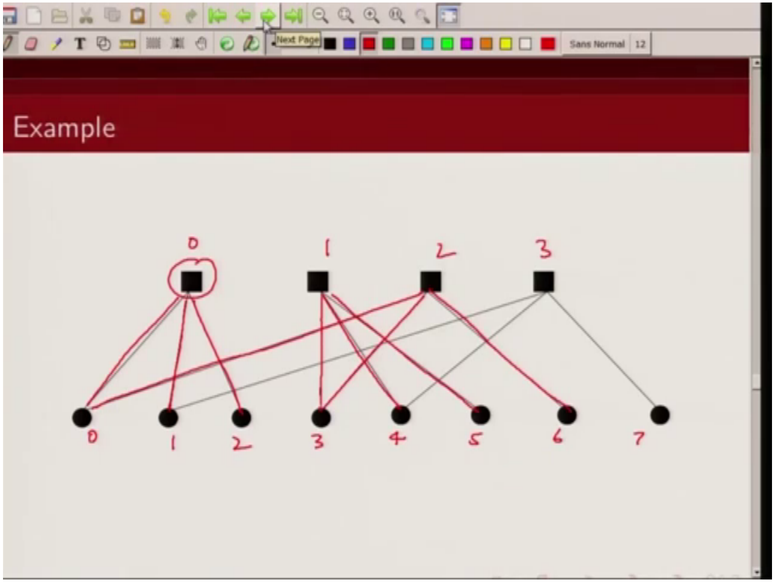
- $\boxed{q_{i,j}(x)}$ is the message passed from the bit node $\underline{X_i = x}$ to the $\underline{j^{th}}$ check node.

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

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

so q i j is the message passed from the ith node to the jth check node, right. So q i js you need to compute for, in this particular example so

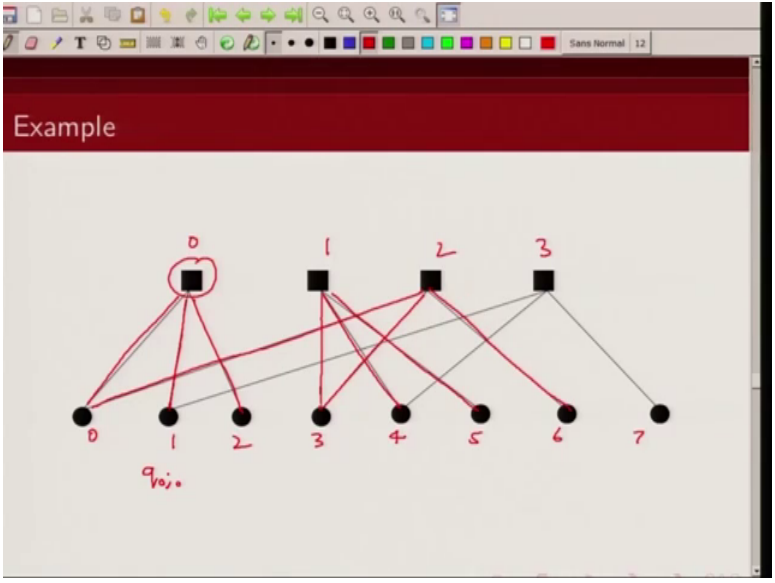
(Refer Slide Time 33:54)



Example

let's look at 0th node. So you need to compute q 0 0 because your message

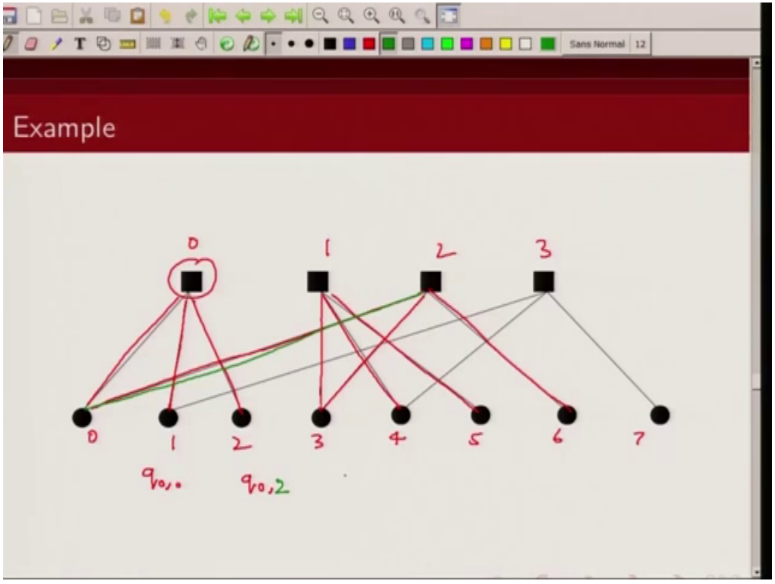
(Refer Slide Time 33:59)



Example

you are sending from 0th node to 0th parity check. You need to compute q 0 2, because you are sending message from 0th bit to the second this thing.

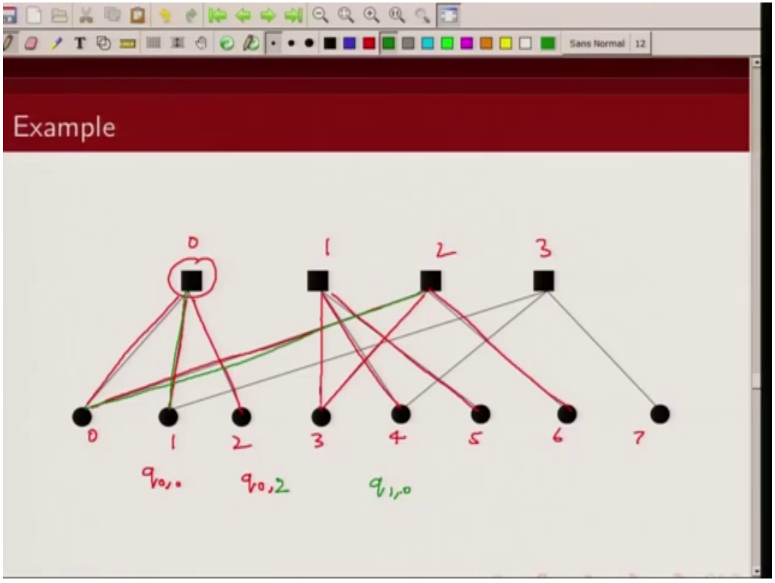
(Refer Slide Time 34:15)



Example

You need to compute this, q 1 0,

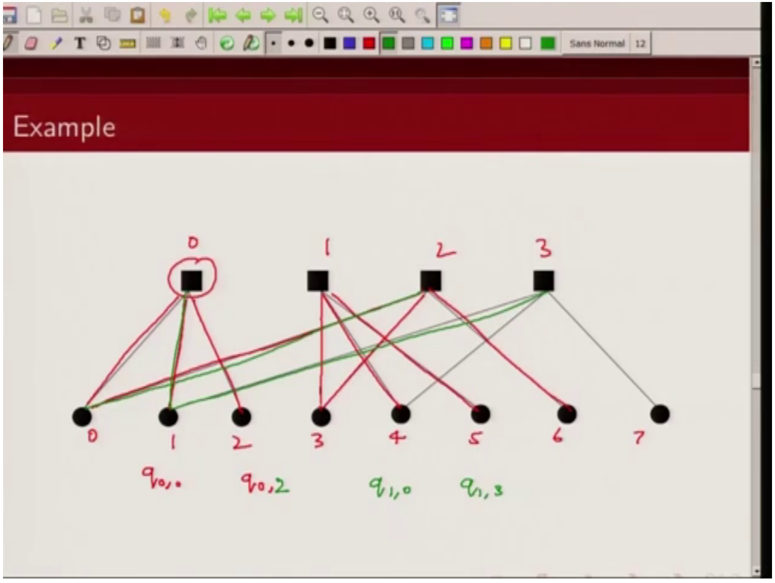
(Refer Slide Time 34:25)



Example

you need to compute q 1 3,

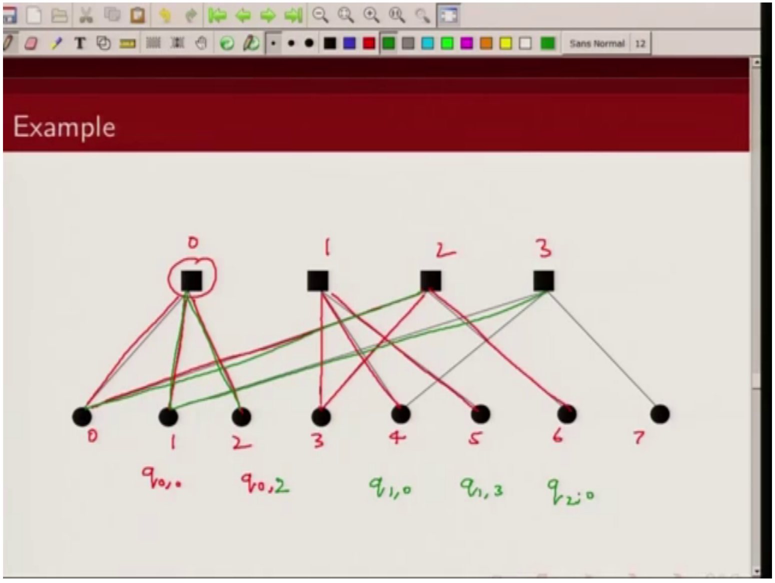
(Refer Slide Time 34:32)



Example

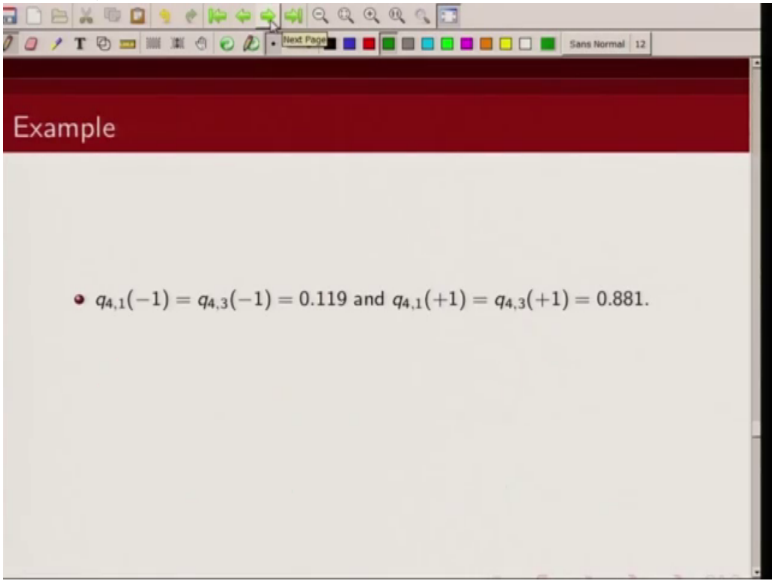
right? You need to compute q 2 0.

(Refer Slide Time 34:39)


Example

And how are these initially computed? These are initially computed based on what are the received y i values that we have to see. And that is precisely what I have shown here. So I have,

(Refer Slide Time 34:55)

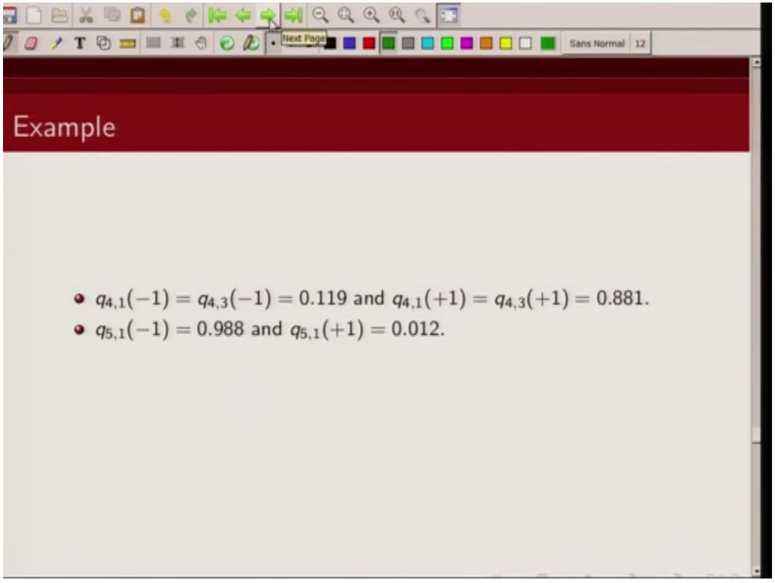


Example

- $q_{4,1}(-1) = q_{4,3}(-1) = 0.119$ and $q_{4,1}(+1) = q_{4,3}(+1) = 0.881$.

you can see here, these are the computed

(Refer Slide Time 34:58)

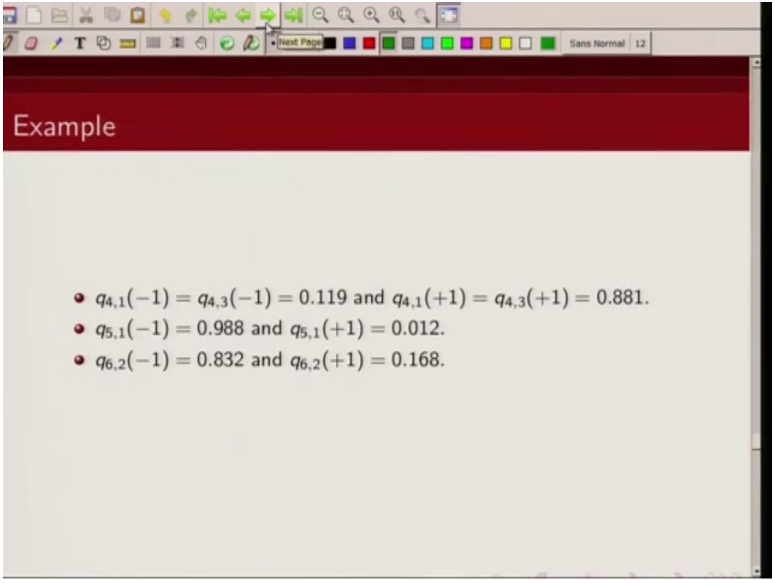


Example

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

values of q is

(Refer Slide Time 35:00)

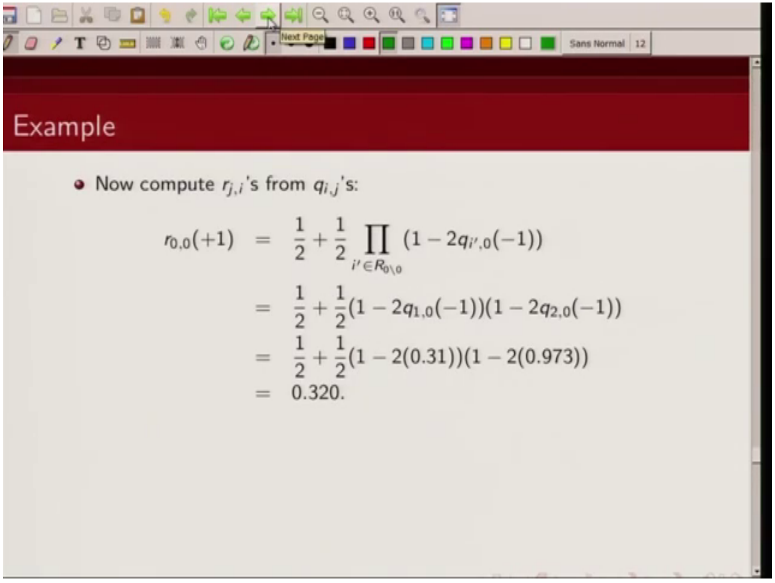


Example

- $q_{4,1}(-1) = q_{4,3}(-1) = 0.119$ 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 $q_{6,2}(+1) = 0.168$.

and pay close attention to these indices. This denotes that ith bit is sending information to the jth parity check constraint, Ok.

(Refer Slide Time 35:18)



Example

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

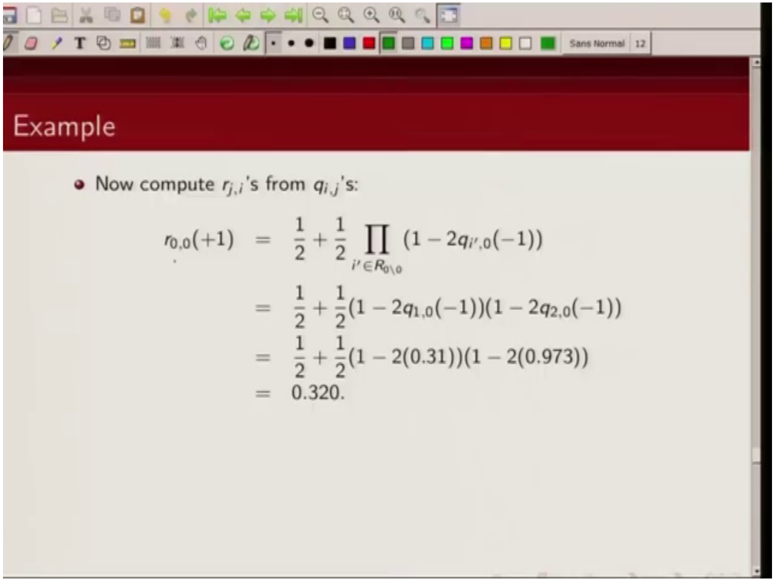
$$\begin{aligned} r_{0,0}(+1) &= \frac{1}{2}+\frac{1}{2}\prod_{i'\in R_{0\backslash 0}}(1-2q_{i',0}(-1)) \\ &= \frac{1}{2}+\frac{1}{2}(1-2q_{1,0}(-1))(1-2q_{2,0}(-1)) \\ &= \frac{1}{2}+\frac{1}{2}(1-2(0.31))(1-2(0.973)) \\ &= 0.320. \end{aligned}$$

So once we have computed this q i j,

(Refer Slide Time 35:21)

then the next thing we need to do is these parity check constraints are now going to check, Ok given that the bit is zero, what is the probability that the parity check constraint is satisfied. Given the bit is 1, what is the probability that parity check constraint is satisfied? And those are

(Refer Slide Time 35:43)



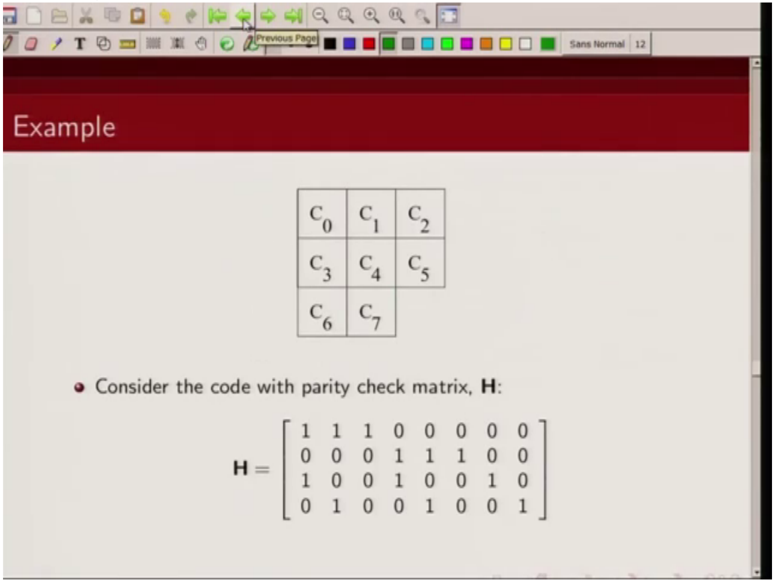
Example

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

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

given by these R's, R ijs right. What is the probability that 0th parity check constraint is satisfied given the 0th bit is plus 1? Now what is that probability? That probability is given by, so we are looking, first parity check constraint now given that first bit's 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's look at the H matrix for the first row.

(Refer Slide Time 36:27)



Example

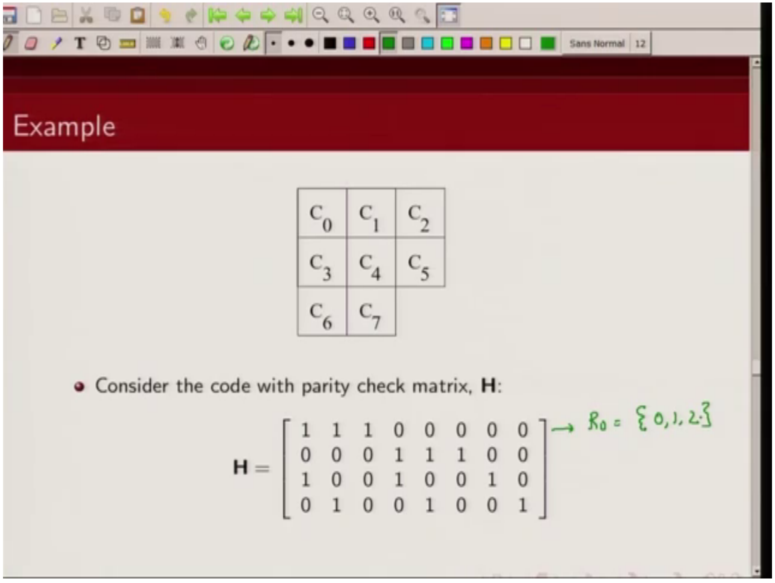
c_0	c_1	c_2
c_3	c_4	c_5
c_6	c_7	

- 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 this is R 0. R 0 is 0, 1 and 2. So

(Refer Slide Time 36:35)



Example

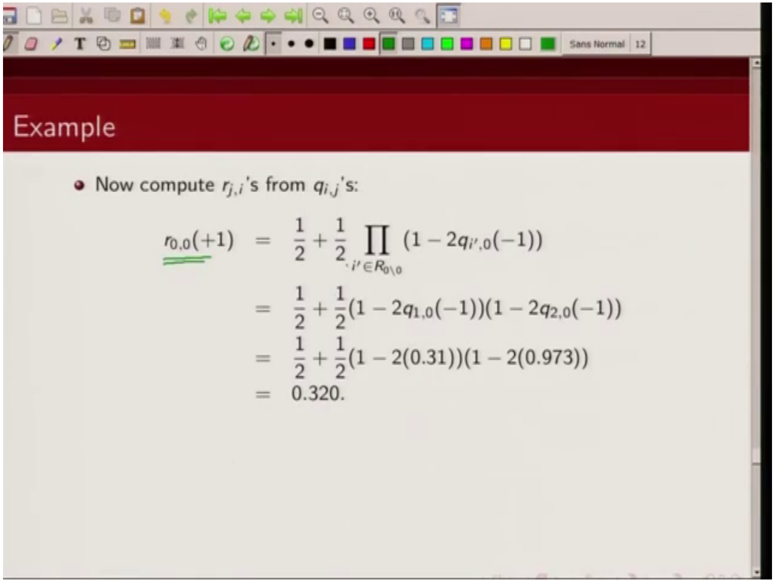
c_0	c_1	c_2
c_3	c_4	c_5
c_6	c_7	

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

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

given that this bit is zero, what we want to find out is what is the probability that sum of these two add up to even parity? And that's what we are doing here.

(Refer Slide Time 36:50)



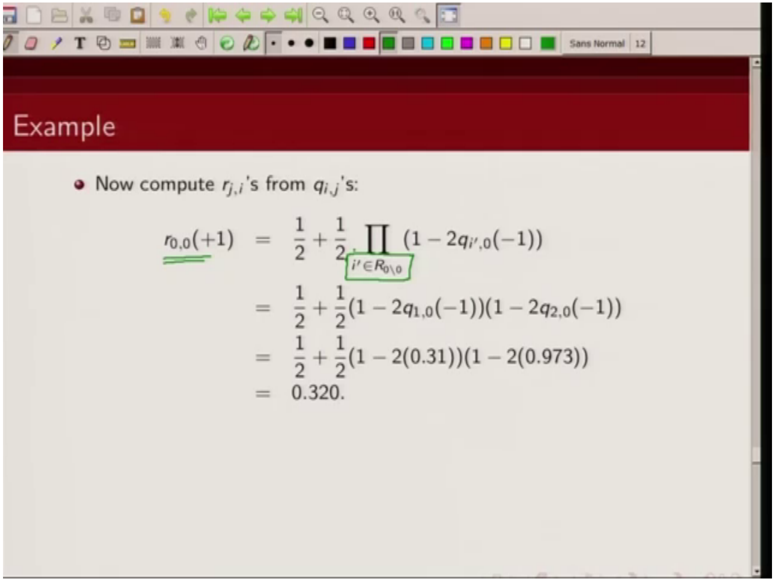
Example

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

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

Note here, ith

(Refer Slide Time 36:54)



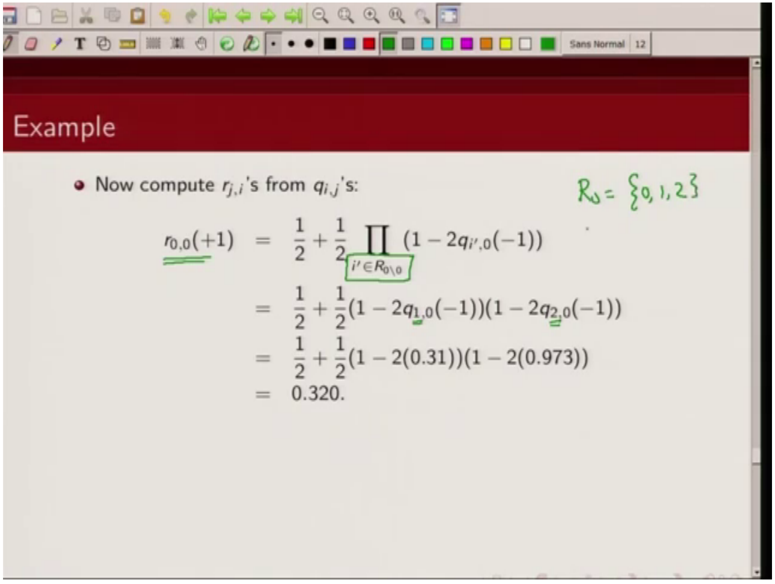
Example

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

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

belongs to R 0 minus this 0th term. So here you will have two terms, one is corresponding to i being 1, and other i being 2. Why, because R 0 was 0,1, 2. So

(Refer Slide Time 37:13)



Example

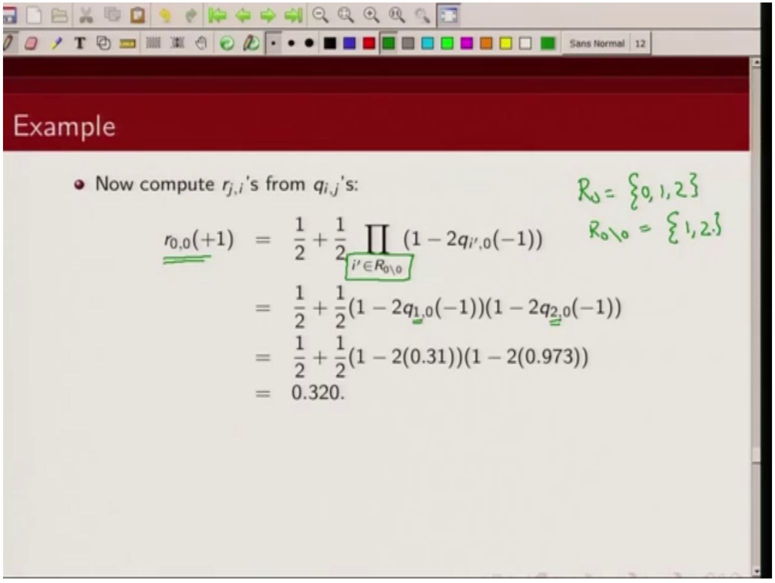
- Now compute $r_{j,i}$'s from $q_{i,j}$'s:

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

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

R 0 minus this element 0 is nothing but 1 and 2.

(Refer Slide Time 37:19)



Example

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

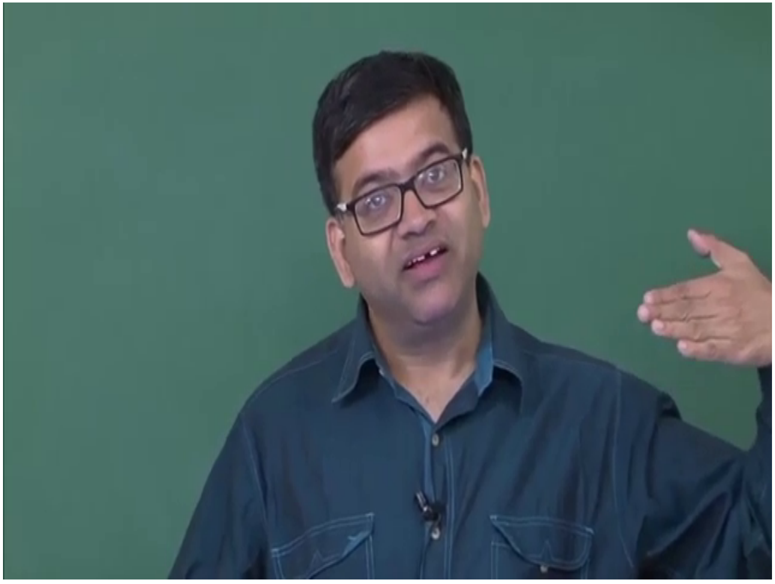
$$\begin{aligned} \underline{r_{0,0}}(+1) &= \frac{1}{2} + \frac{1}{2} \prod_{\boxed{i' \in R_{0\setminus 0}}} (1 - 2q_{i',0}(-1)) \\ &= \frac{1}{2} + \frac{1}{2}(1 - 2q_{\underline{1},0}(-1))(1 - 2q_{\underline{2},0}(-1)) \\ &= \frac{1}{2} + \frac{1}{2}(1 - 2(0.31))(1 - 2(0.973)) \\ &= 0.320. \end{aligned}$$

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

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

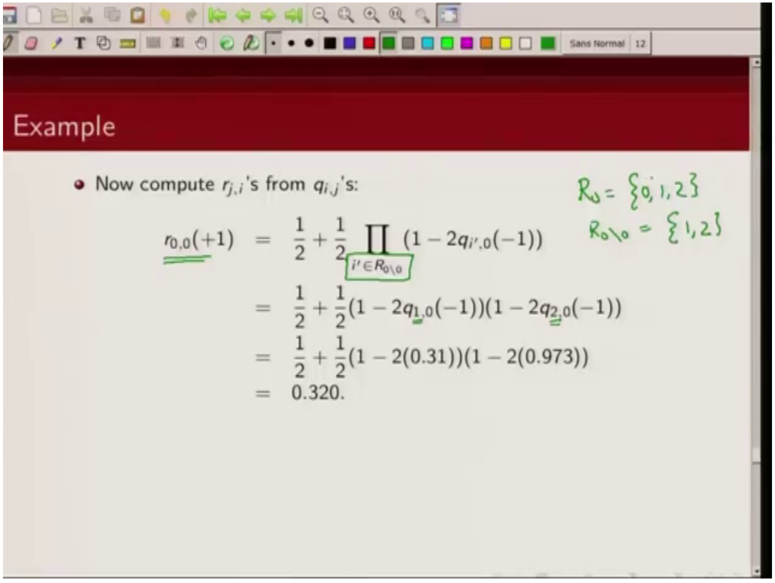
So here in this product you will have 2 terms, one corresponding to q 1 0, other corresponding to q 2 0. Because in that

(Refer Slide Time 37:29)



0th parity check equation, other than the first bit, the bits that are participating is that, other than the 0th bit, other

(Refer Slide Time 37:37)



Example

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

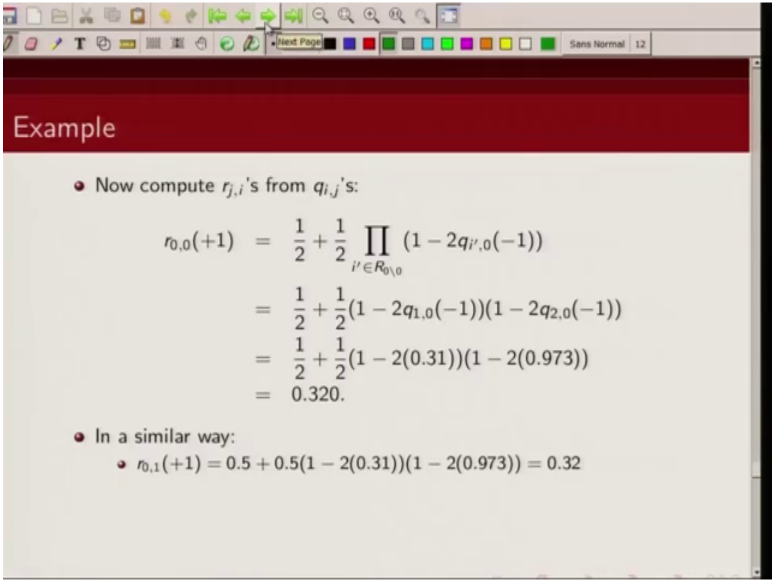
$$\begin{aligned} \underline{\underline{r_{0,0}}}(+1) &= \frac{1}{2}+\frac{1}{2}\prod_{\boxed{i'\in R_{0\backslash 0}}}(1-2q_{i',0}(-1)) \\ &= \frac{1}{2}+\frac{1}{2}(1-2q_{\underline{1},0}(-1))(1-2q_{\underline{\underline{2}},0}(-1)) \\ &= \frac{1}{2}+\frac{1}{2}(1-2(0.31))(1-2(0.973)) \\ &= 0.320.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

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

$R_{0\backslash 0} = \{1,2\}$

bits which are participating is bit number 1 and bit number 2. So this is the probability that parity check constraint is satisfied given c i is 0 and similarly we can calculate the other

(Refer Slide Time 37:52)



Example

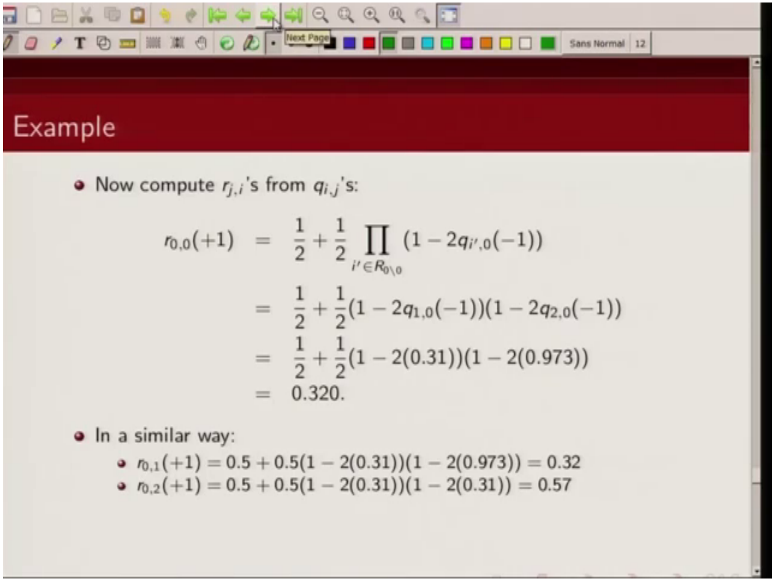
- Now compute $r_{j,i}$'s from $q_{i,j}$'s:

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

- In a similar way:
  - $r_{0,1}(+1) = 0.5 + 0.5(1-2(0.31))(1-2(0.973)) = 0.32$

R is. I am not going

(Refer Slide Time 37:54)



Example

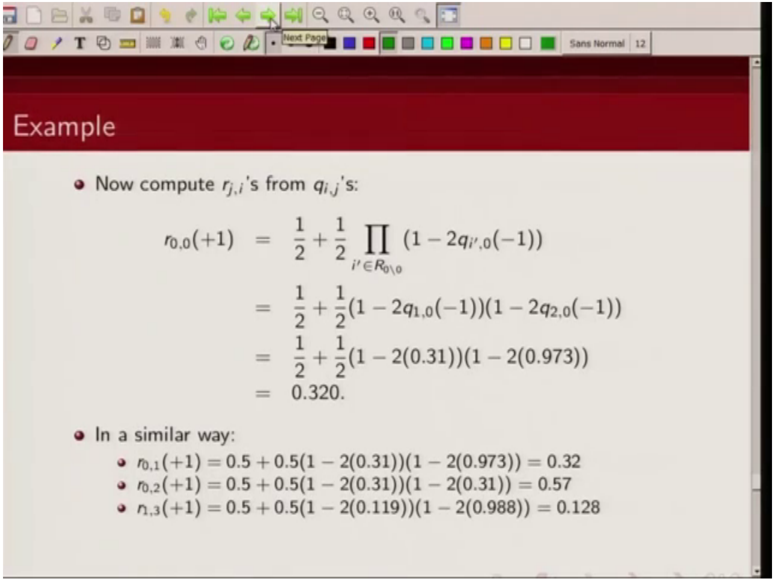
- Now compute $r_{j,i}$'s from $q_{i,j}$'s:

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

- In a similar way:
    - $r_{0,1}(+1) = 0.5 + 0.5(1 - 2(0.31))(1 - 2(0.973)) = 0.32$
    - $r_{0,2}(+1) = 0.5 + 0.5(1 - 2(0.31))(1 - 2(0.31)) = 0.57$

into detail of that. It’s just the same procedure repeated. So once we calculate these

(Refer Slide Time 38:00)



Example

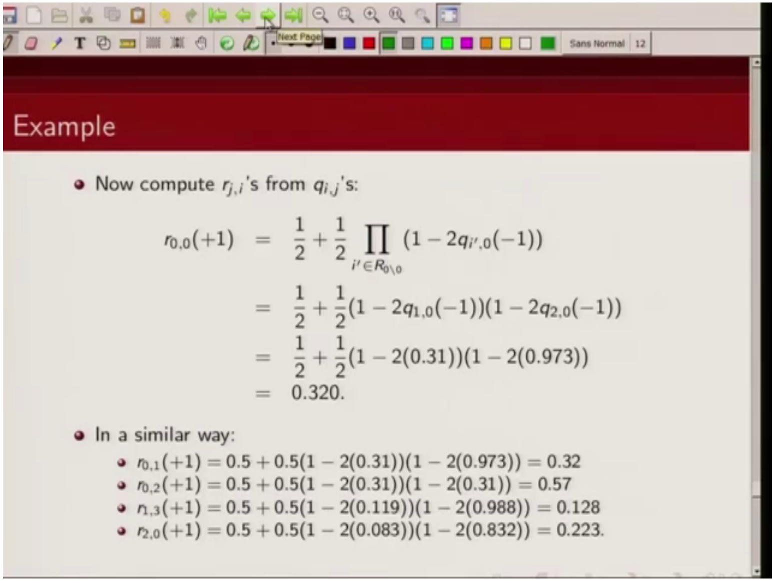
- Now compute $r_{j,i}$'s from $q_{i,j}$'s:

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

- In a similar way:
    - $r_{0,1}(+1) = 0.5 + 0.5(1 - 2(0.31))(1 - 2(0.973)) = 0.32$
    - $r_{0,2}(+1) = 0.5 + 0.5(1 - 2(0.31))(1 - 2(0.31)) = 0.57$
    - $r_{1,3}(+1) = 0.5 + 0.5(1 - 2(0.119))(1 - 2(0.988)) = 0.128$

q is,

(Refer Slide Time 38:01)



Example

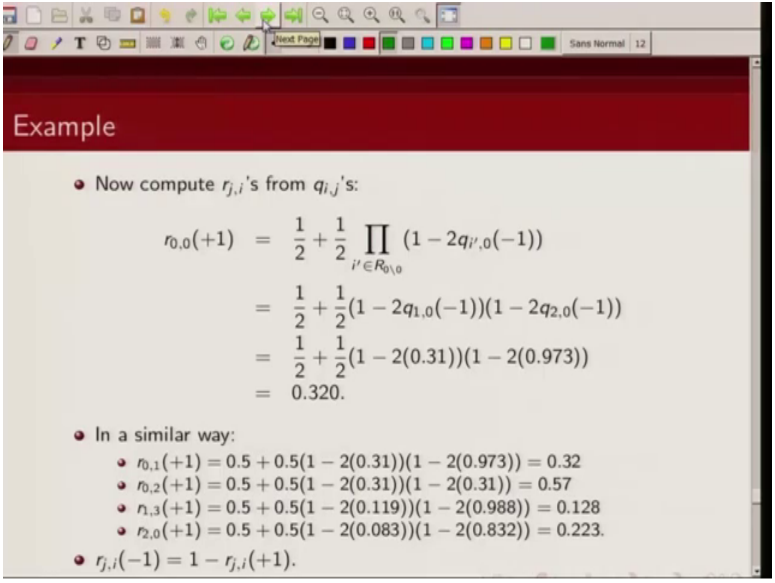
- Now compute $r_{j,i}$'s from $q_{i,j}$'s:

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

- In a similar way:
  - $r_{0,1}(+1) = 0.5+0.5(1-2(0.31))(1-2(0.973)) = 0.32$
  - $r_{0,2}(+1) = 0.5+0.5(1-2(0.31))(1-2(0.31)) = 0.57$
  - $r_{1,3}(+1) = 0.5+0.5(1-2(0.119))(1-2(0.988)) = 0.128$
  - $r_{2,0}(+1) = 0.5+0.5(1-2(0.083))(1-2(0.832)) = 0.223.$

R is and initial

(Refer Slide Time 38:03)



Example

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

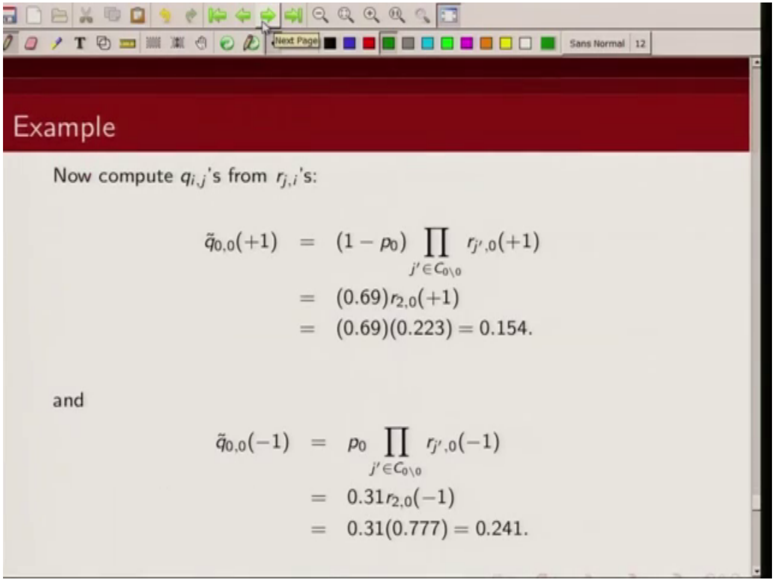
$$\begin{aligned}
r_{0,0}(+1) &= \frac{1}{2}+\frac{1}{2}\prod_{i'\in R_{0\setminus 0}}(1-2q_{i',0}(-1)) \\
&= \frac{1}{2}+\frac{1}{2}(1-2q_{1,0}(-1))(1-2q_{2,0}(-1)) \\
&= \frac{1}{2}+\frac{1}{2}(1-2(0.31))(1-2(0.973)) \\
&= 0.320.
\end{aligned}$$

- In a similar way:
  - $r_{0,1}(+1) = 0.5+0.5(1-2(0.31))(1-2(0.973)) = 0.32$
  - $r_{0,2}(+1) = 0.5+0.5(1-2(0.31))(1-2(0.31)) = 0.57$
  - $r_{1,3}(+1) = 0.5+0.5(1-2(0.119))(1-2(0.988)) = 0.128$
  - $r_{2,0}(+1) = 0.5+0.5(1-2(0.083))(1-2(0.832)) = 0.223.$

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

q is, then we are going to

(Refer Slide Time 38:05)



Example

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

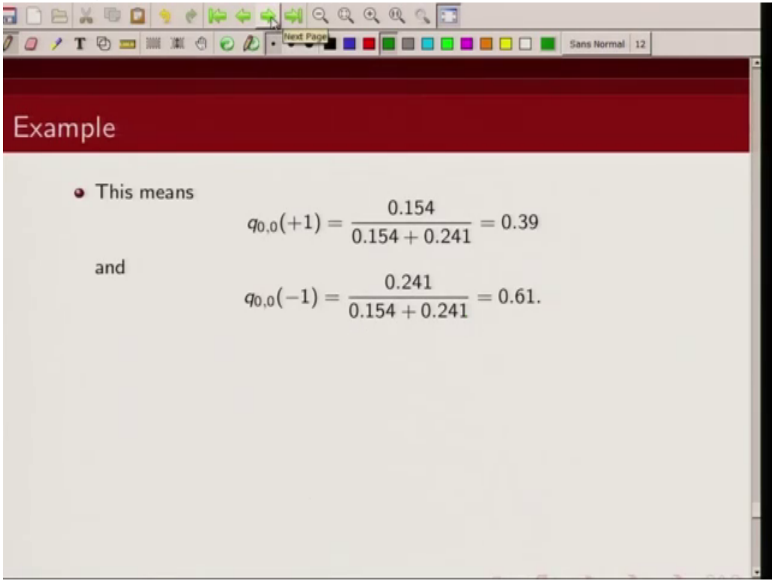
$$\begin{aligned}\tilde{q}_{0,0}(+1) &= (1-p_0)\prod_{j'\in C_{0\backslash 0}} r_{j',0}(+1)\\ &= (0.69)r_{2,0}(+1)\\ &= (0.69)(0.223) = 0.154.\end{aligned}$$

and

$$\begin{aligned}\tilde{q}_{0,0}(-1) &= p_0\prod_{j'\in C_{0\backslash 0}} r_{j',0}(-1)\\ &= 0.31r_{2,0}(-1)\\ &= 0.31(0.777) = 0.241.\end{aligned}$$

update our q is. 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 plus 1 and minus 1 and we can normalize these probabilities so that they sum up to

(Refer Slide Time 38:35)



Example

- This means

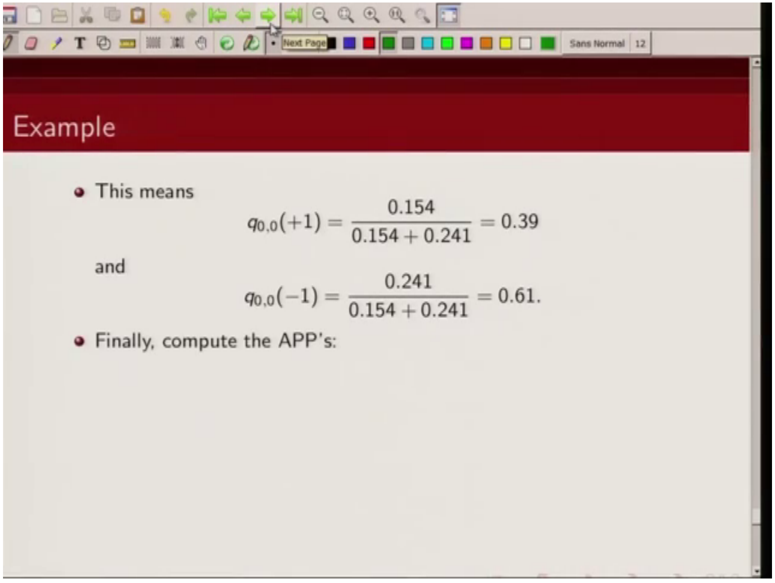
$$q_{0,0}(+1) = \frac{0.154}{0.154+0.241} = 0.39$$

and

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

1 so if we normalize it, these are the probabilities we are getting, right and next thing,

(Refer Slide Time 38:43)



Example

- This means

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

and

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

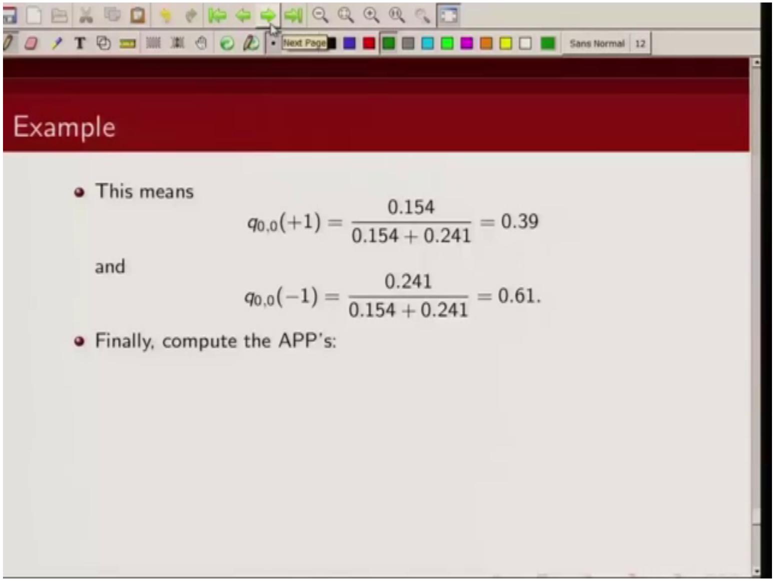
- Finally, compute the APP's:

finally after we have computed one round of iteration; what is one round of iteration,

(Refer Slide Time 38:49)

you send these q is to check node then check node gives you back R is, y update your q is

(Refer Slide Time 38:59)



Example

- This means

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

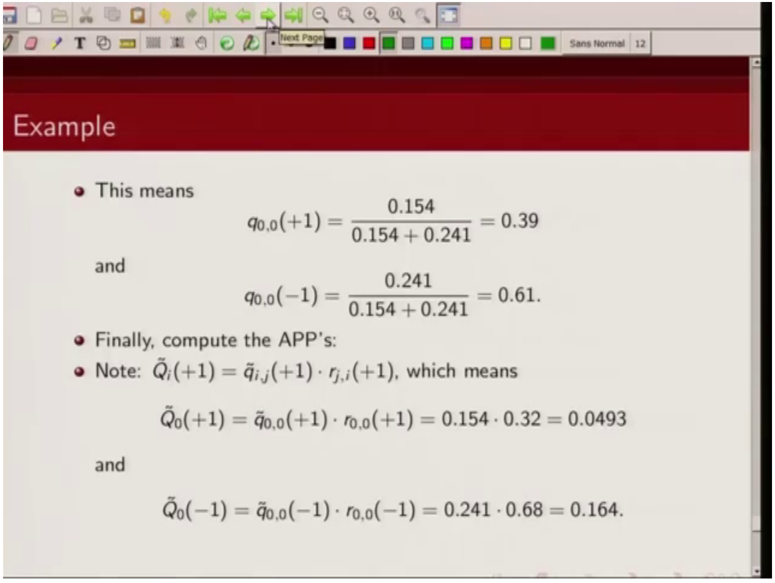
and

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

- Finally, compute the APP's:

and once you do that,

(Refer Slide Time 39:01)



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:
- Note: $\tilde{Q}_i(+1) = \tilde{q}_{i,j}(+1) \cdot r_{j,i}(+1)$, which means

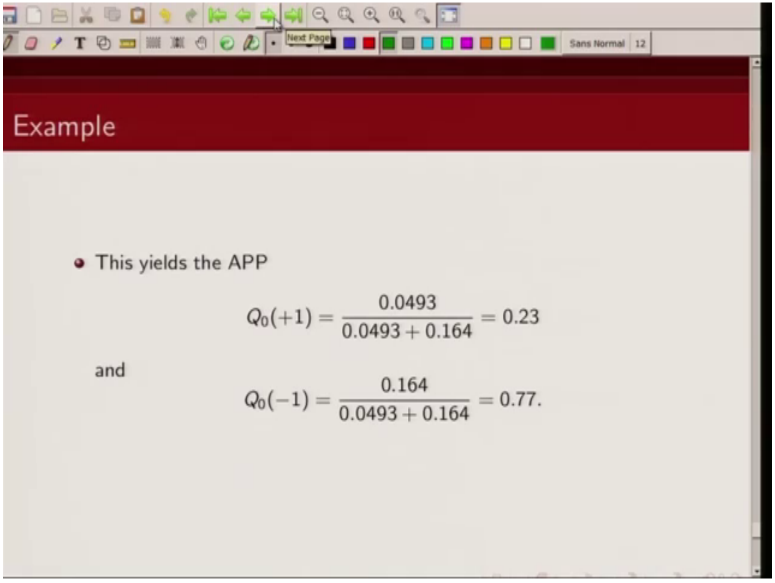
$$\tilde{Q}_0(+1) = \tilde{q}_{0,0}(+1) \cdot r_{0,0}(+1) = 0.154 \cdot 0.32 = 0.0493$$

and

$$\tilde{Q}_0(-1) = \tilde{q}_{0,0}(-1) \cdot r_{0,0}(-1) = 0.241 \cdot 0.68 = 0.164.$$

you can find out the a posteriori probability which we have done here. So probability of it being 1, probability of this being 0, and then based on, again you can normalize these

(Refer Slide Time 39:14)



Example

- This yields the APP

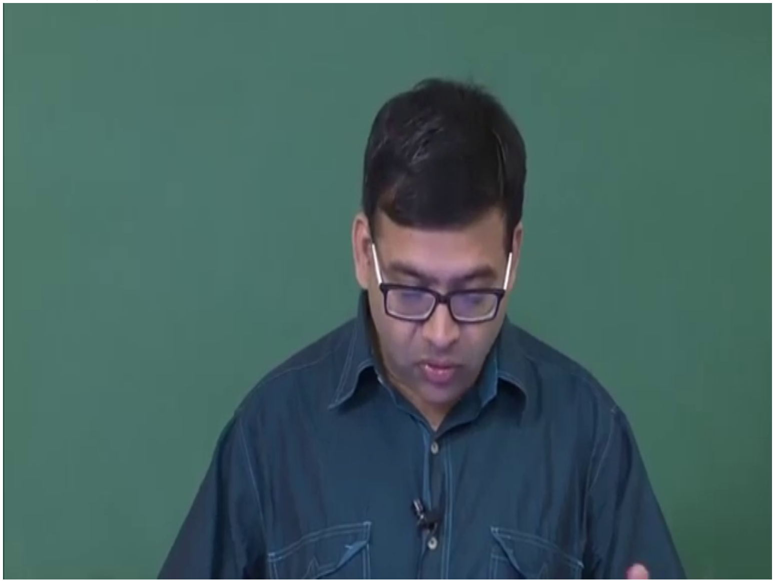
$$Q_0(+1) = \frac{0.0493}{0.0493 + 0.164} = 0.23$$

and

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

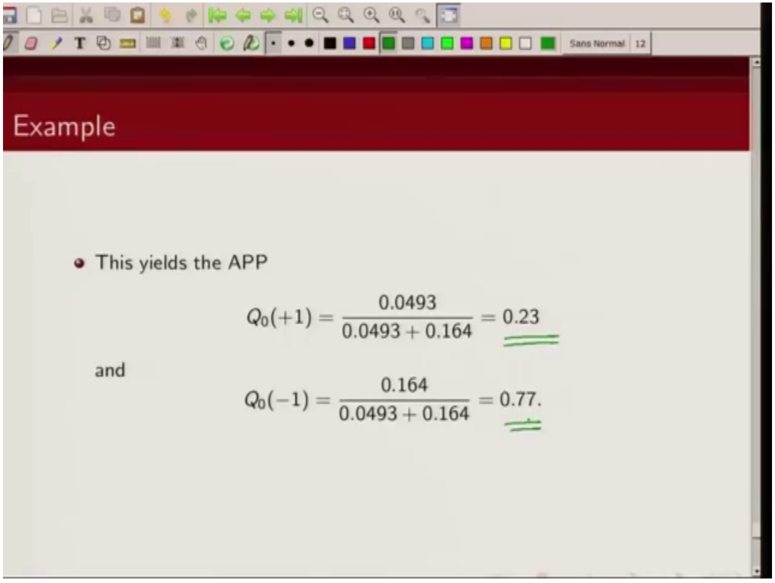
probabilities so that comes out to be this. Now based on whichever is more likely you decide in favor of that. For example, in this case q being minus 1 is more likely so the bit which was transmitted was 1, Ok.

(Refer Slide Time 39:31)



So to recap how this probabilistic decoding algorithm works, you have your received sequence. You compute these probabilities q is, message bits, send it to the check bits, the check bits then do some compute, local computation, sends the information back. Now this process goes on and on until basically all the parity checks are

(Refer Slide Time 39:59)



Example

- This yields the APP

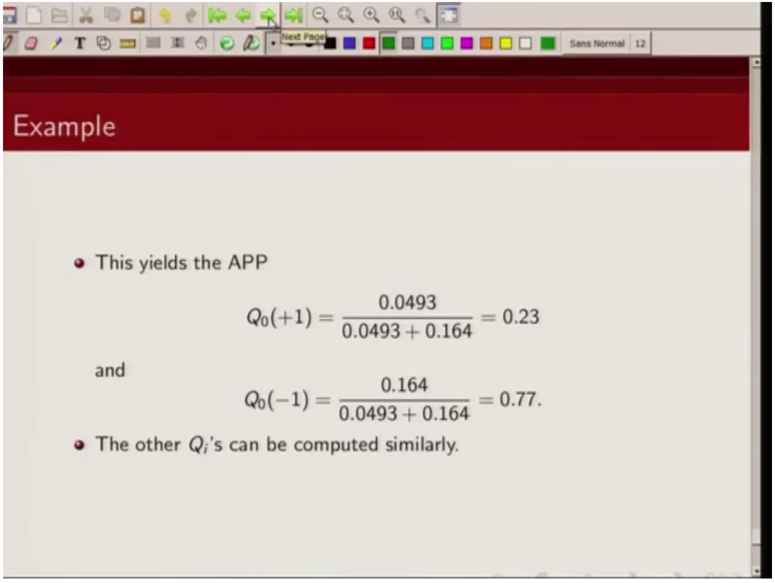
$$Q_0(+1) = \frac{0.0493}{0.0493 + 0.164} = \underline{\underline{0.23}}$$

and

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

satisfied. Or the maximum number of iterations are

(Refer Slide Time 40:01)



Example

- This yields the APP

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

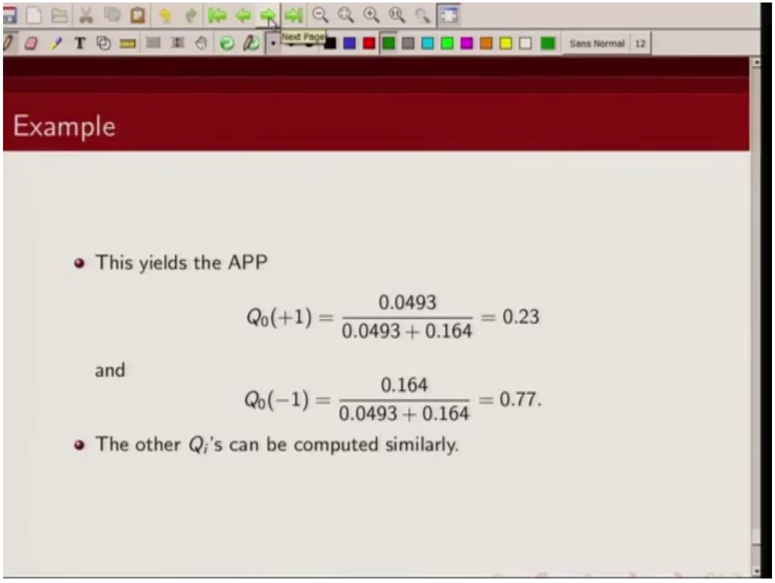
and

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

- The other Q_i's can be computed similarly.

reached.

(Refer Slide Time 40:02)



Example

- This yields the APP

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

and

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

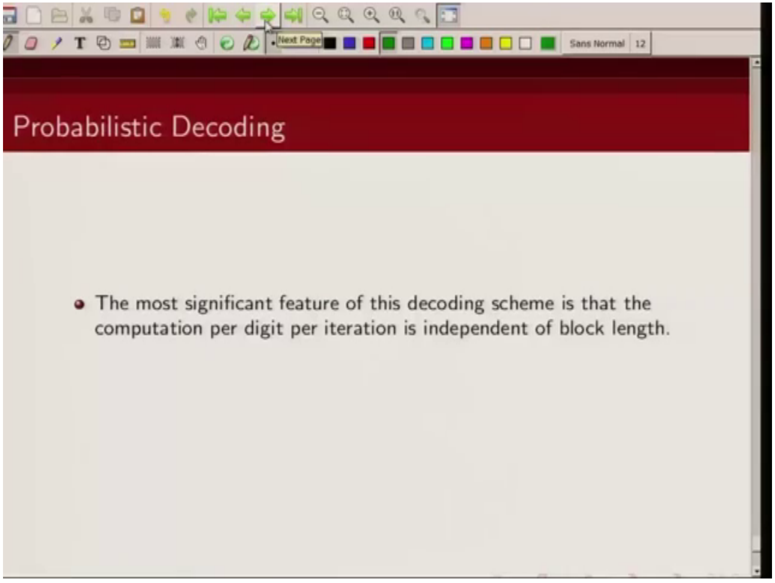
- The other Q_i's can be computed similarly.

Now this we have done for the first bit.

(Refer Slide Time 40:04)

You can do the same thing for other bits as well. In fact the beauty of this algorithm is that you can do this whole operation parallely. So one of

(Refer Slide Time 40:15)

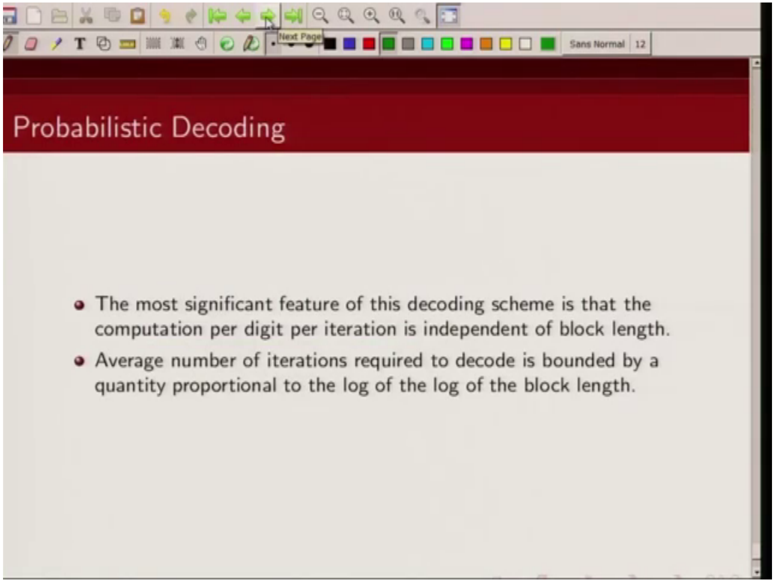


Probabilistic Decoding

- The most significant feature of this decoding scheme is that the computation per digit per iteration is independent of block length.

the nicest features of this algorithm is computation per bit per digit is independent of clock size. And

(Refer Slide Time 40:23)



Probabilistic Decoding

- The most significant feature of this decoding scheme is that the computation per digit per iteration is independent of block length.
- Average number of iterations required to decode is bounded by a quantity proportional to the log of the log of the block length.

average number of iterations required basically to decode is typically bounded by log log of n. After that you started getting correlated information back. So with this, we will conclude our discussion on

(Refer Slide Time 40:39)

probabilistic decoding of L D P C codes, thank you.