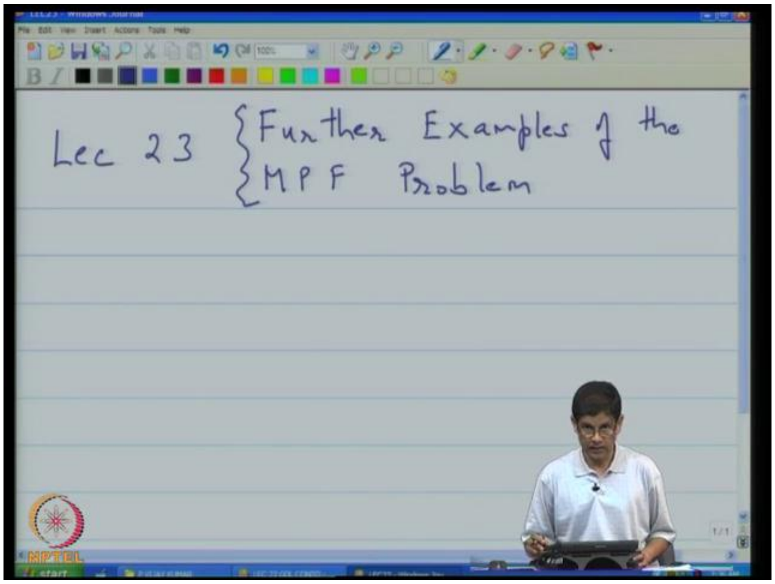
Error Correcting Codes
Prof. Dr. P Vijay Kumar
Department of Electrical Communication Engineering
Indian Institute of Science, Bangalore

Lecture No. # 23
Further Examples of the MPF Problem

(Refer Slide Time: 00:29)

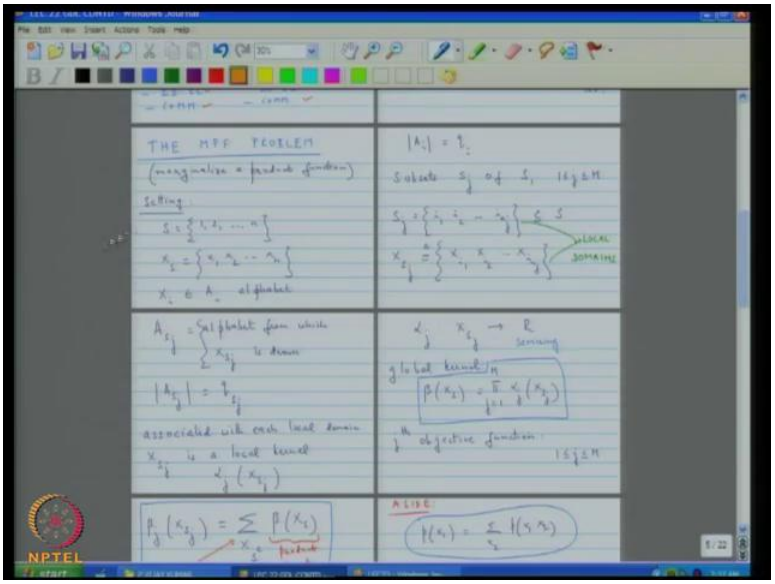


Good afternoon and welcome back, let me begin with usual recap. So, this will be lecture 23, and I will call this I use the title, further examples of the MPF problem. And let us just have a quick overview of discussion in the last lecture; in the last lecture, we are looking, we finished concluded our discussion on semi ring. We looked at max product and min sum semi ring, and then I introduced; so where we headed is that we are trying to give, what you might call some mathematical underpinnings for which are the basis of couple of decoding algorithms that are popularly used today. And I am following a certain perspective, so different people have different approaches; I am following a certain perspective from certain people. And I will mention this people little later in the lecture series. So, from that perspective, we actually introduced the notion of a semi ring.

We give examples of semi ring, and then as we will see later, you can actually formulate the problem of decoding into the problem of an MPF problem. An MPF stands for Marginalize

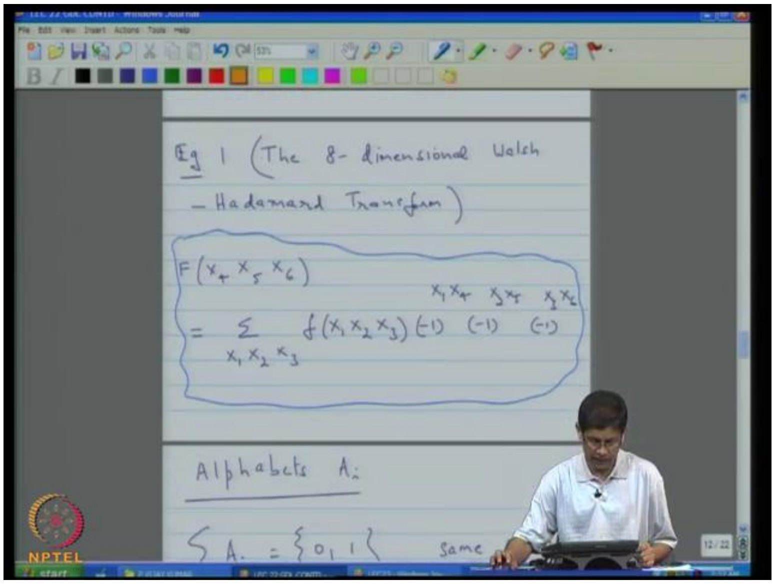
Product Function; and we were looking at certain first I will give you general setting of marginalize product function problem; there are universal set local domains.

(Refer Slide Time: 02:24)



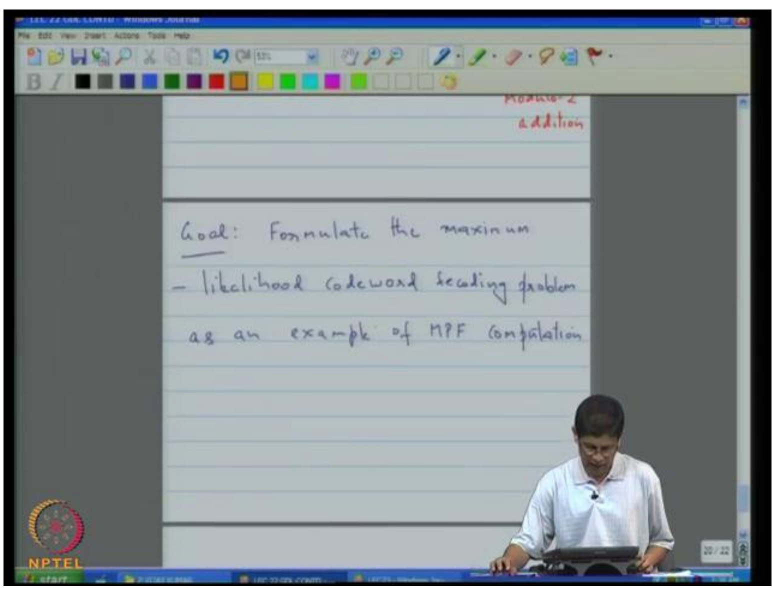
So, there are there is universal set as there are subset is known in local domains, each subset is associated to a local kernel. Then there is global kernel, which is the product of the local kernels, and eventually what you are interested is actually is in marginalizing the product function with respect to certain subset of variables.

(Refer Slide Time: 02:58)



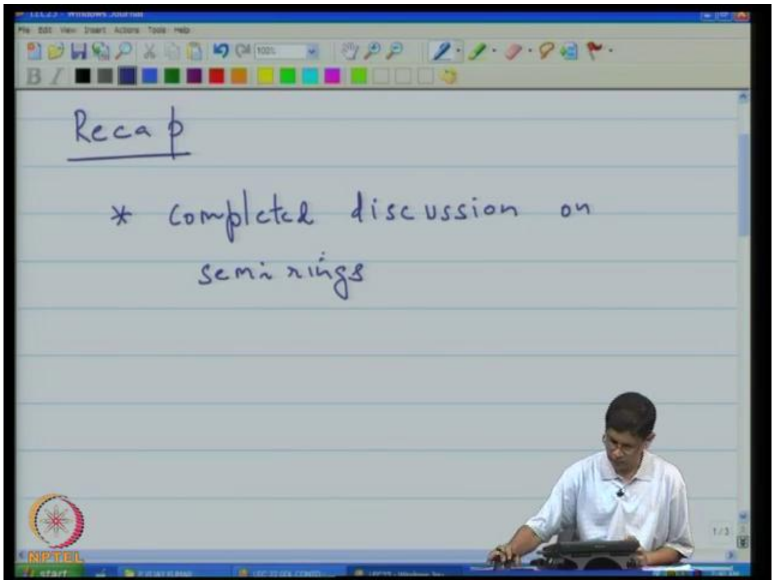
Now I as a first example, I give you an example of a kind of transform. So, this transform is similar to Fourier transform is called Walsh transform. And I give this example, and I showed you how one can view it as an example of an MPF problem.

(Refer Slide Time: 03:29)



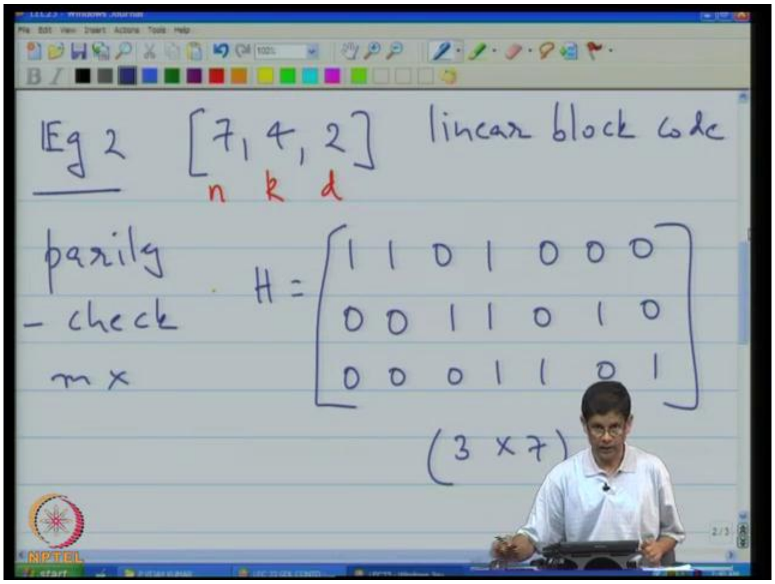
And towards under the lecture, I started discussing as an example of decoding of a particular block code, and the block code is actually shown on this piece of paper. So, what I like to do is I like to copy over this particular page. Our goal is that we are going consider decoding of this code over the binary symmetric channel. And the goal is then to formulate maximum like to code word decoding of this code as an example of MPF computation.

(Refer Slide Time: 04:38)



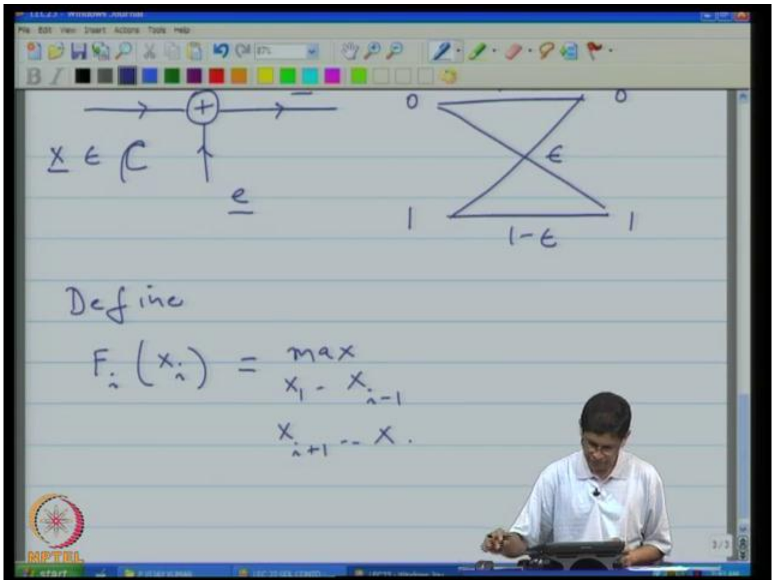
Let us put down a usual recap complete discussion on semi rings. And I defined the MPF problem and we started looking at examples of the MPF problem. The first example, we look up that of the fast Walsh transform.

(Refer Slide Time: 05:44)



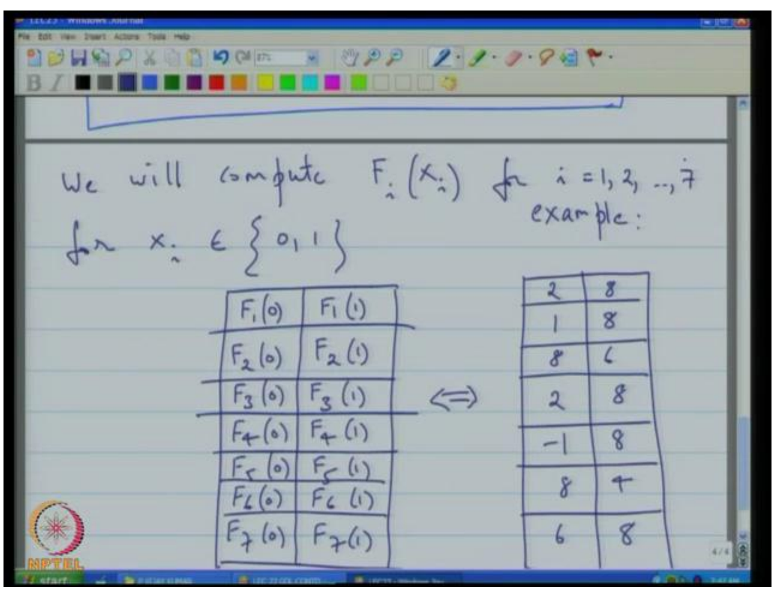
So, the second example is of this particular linear block code. So, this is the code that has block length seven and the dimension is four, and the minimum distance is two. The code is described in terms of parity check matrix and the parity check matrix is actually shown here. They are interested in maximum likelihood code word decoding. Let us make that clear, our interest is in maximum likelihood code word decoding of this block code. And let me do this, I will write down some equations and then I will come back explain how this equation can be used to solve problem of maximum likelihood code word decoding.

(Refer Slide Time: 07:12)



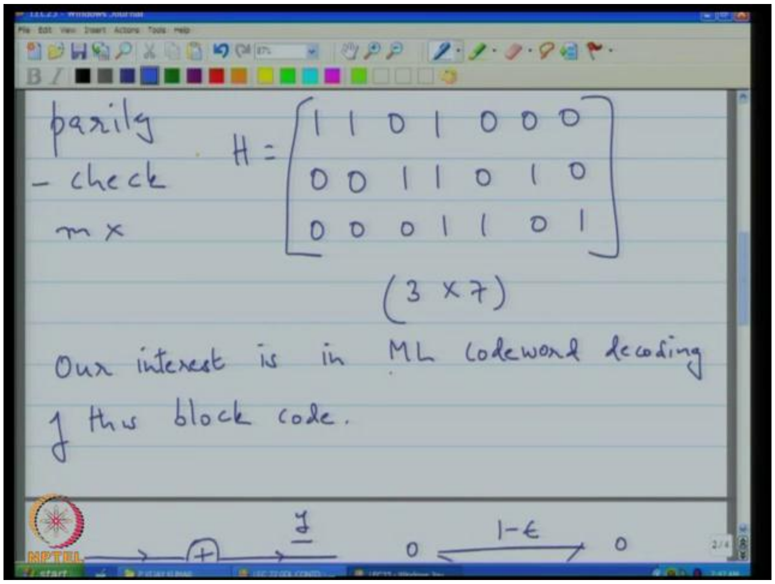
So, we are going to decode this over bandwidth symmetric channel. Whether input is X and is drawn from the code word, the channel introduce in error vector and output of the channel is Y. Defined F i of X i to be the max over X 1 to X i minus 1, X i plus 1 to X 7. Further subject to the restriction that X belongs to the code and then we have P of Y given X.

(Refer Slide Time: 09:22)



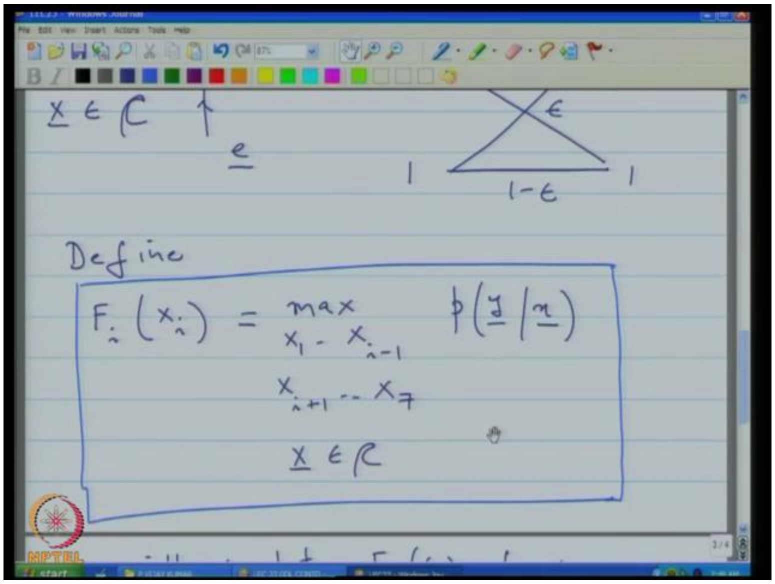
So, we look at this function, so we will compute F i (X i) for i equal to 1, 2, to 7, for X i in 0 and 1. And then you might think of the next step has being the writing (( )) of table, where we put F 1 of 0, F 2 of 0, F 3 of 0 and so on. And let me give you an example of what one might encounter, then one compute such a table. So, when you compute this might look like 2, 8, 1, 8, 6, 2, 8, minus 1, 8, 4 and 6, 8. When you compute this, you might find in example computation that you get this.

(Refer Slide Time: 12:12)



Where my headed with this, I am just saying that or interest is in maximum likelihood code word decoding of this convolutional code and might name is that. If you follow the procedure, then we learn up doing that. And I think, let as assume that particular transmission, there is certain code word is transmitted; X was transmitted and Y was received. And the situation was such that there is particular code word of X for which is P of Y given X is a maximum. Maximum likelihood code word decoding requires you to examine this quantity, and look for that code word X that maximized that. But the problem is that we want to fit this in to the framework of MPF problem. And the one way in which can actually do this define these function F i of X i.

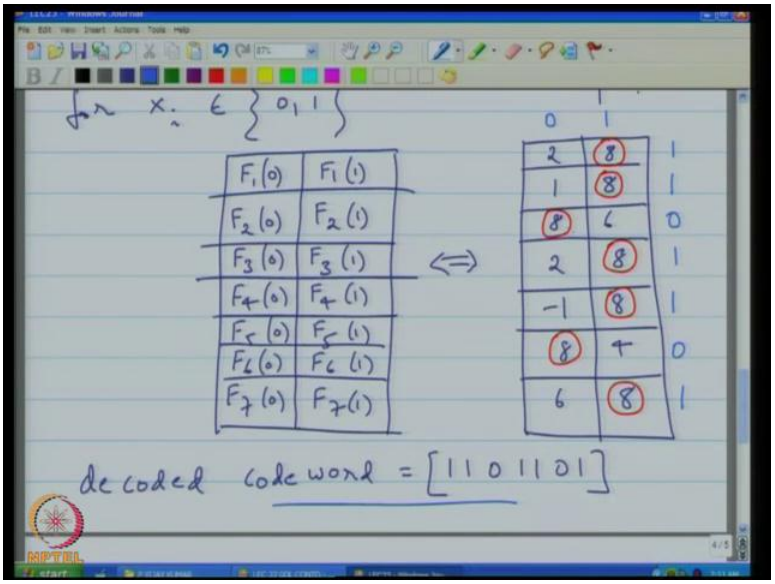
(Refer Slide Time: 13:13)



Now it is easy to understand how this works. If you assume that there is a unique code word X such that is a maximum for a given Y, so we will assume that. Then then you determine F i of X i, what you going to do is you going to examine this quantity for all vectors X. This value of X i with X i is given by here, and the additional constraint that X must belong to the code. Now what will happen to that certainly, when you let say the most likely code word is going to have in the higher position either zero or one, let say it has one.

That means, when you compute this you will also compute this quantity corresponding to X i equal to zero, and X i equal to one. Then you will obviously find in the value computed for X i equal to one is larger, simply because X i equal to one is part of code word is actually maximize this. So, by comparing F i of zero and F i of one and choosing larger one, here recovering component of maximize likelihood code word. So, that is have the actually work.

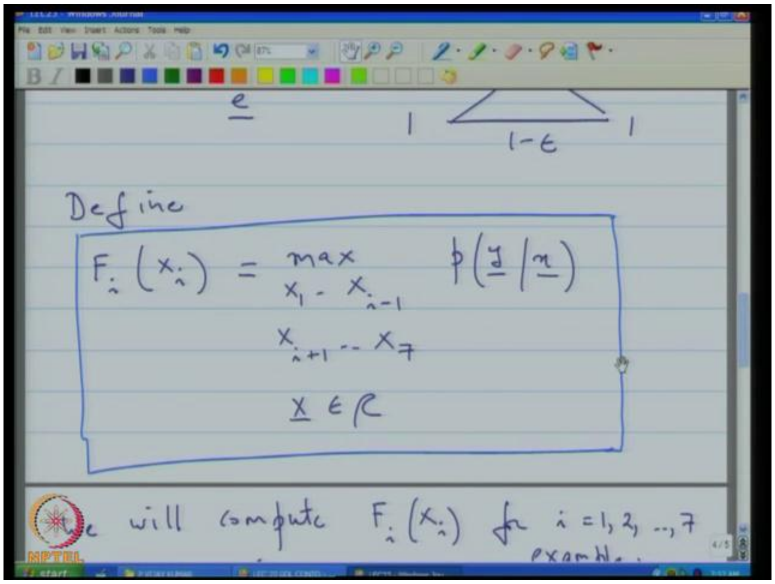
(Refer Slide Time: 14:36)



And this table just helps you visualize this, what you do is compute F 1(0), F 1(1) and then you look here. And in each instance, you actually make a decision which of these two quantities is larger and similarly, you to the same for this and this and so on. An example, you might get something like this, suppose in the most likely code word has the value of this quantity equal to eight. Then you do this computation, you will actually under getting tables which looks something like this. Then what you going to do is you are going to actually pick the larger of the two in all the instances, so which in this case are eight. And then you are going to declare based on this, because this quantity here stands for zero and this for one. That is the most likely code word here has component, which are 1 1 0 1 1 0 1.

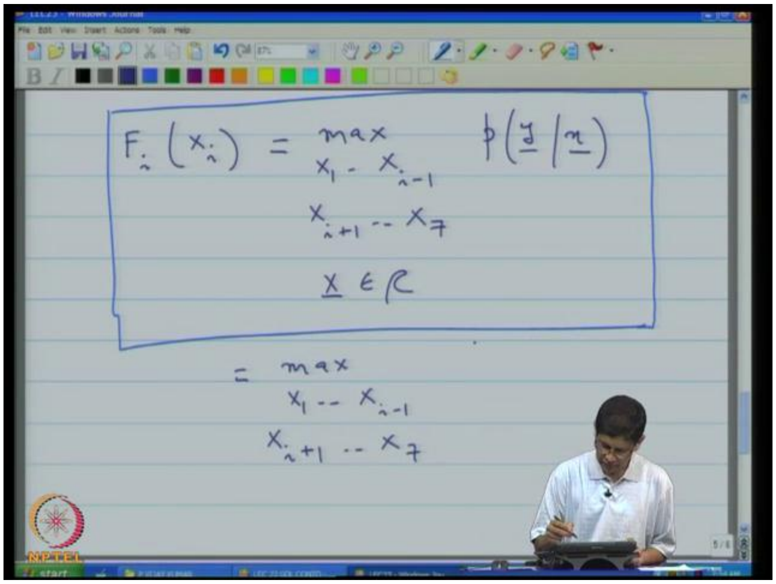
So, this then the vector is here your decoded code word. So, the decoded code word is then 1 1 0 1 1 0 1. Now the reason was you have this quantity each showing up here. Because in this example, what happened was that the code word which there was unique code word that maximizes this. There was the unique code word that maximizes this quantity and that particular maximum happened to be eight. That number eight shows up in every component. Because if you try to maximize this with the fourth F four of X four equal to zero. Then what will happen is that? For X 4 equal to one that code word will be considered in this maximization, and the value eight will show up.

(Refer Slide Time: 17:01)



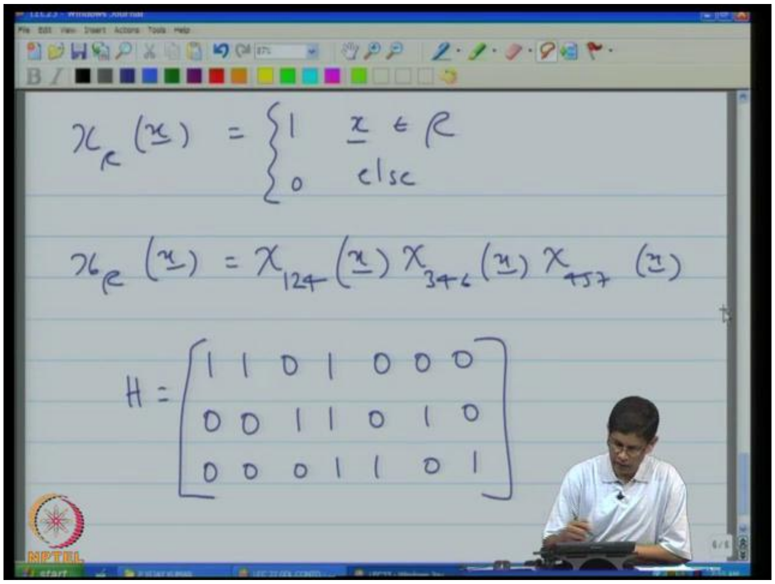
And this knows the value that value be (( )). So, you will the value for X 4 equal to one, therefore greater than the value for X4 equal to zero. You would run up the comparison of eight was two in that particular case for the fourth example. Similarly, will get comparison of in all the other cases and eight is the common value of P of Y given X that will actually decode. So, that is in so for as explaining, why this computation is most likelihood code word? Now what I want to actually show you is something which parts is easier. Now having frame date in this manner will just squared in show, where this is an example of an MPF computation.

(Refer Slide Time: 17:50)



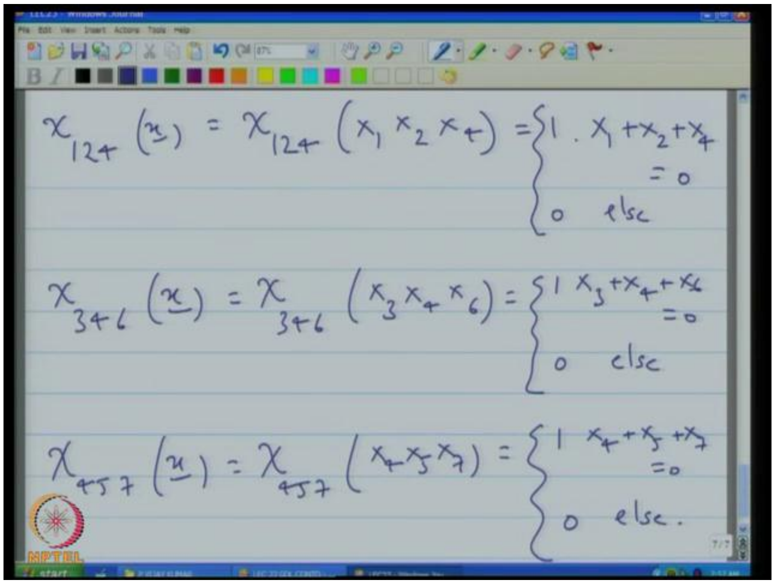
So, I am going to unique copy this expression, because I want to rewrite it slightly differently. And what I want to do is I want to rewrite this K, see that you are carrying maximum all possible vectors X 1 to X 7 excluding the component X i here. But you are writing the restriction that X belongs to the code, so I want to do remove that. So, to give that will rewrite this using a small trick as the max over X 1 to X i minus 1, X i plus 1 to X 7 will remove the restriction that is summation take place over the code. But we will insert it, will take care of that additional play, what will do is will put down P of Y given X. And we will put down a function X, which have will write X c of X; this is script X c of little x.

(Refer Slide Time: 19:27)



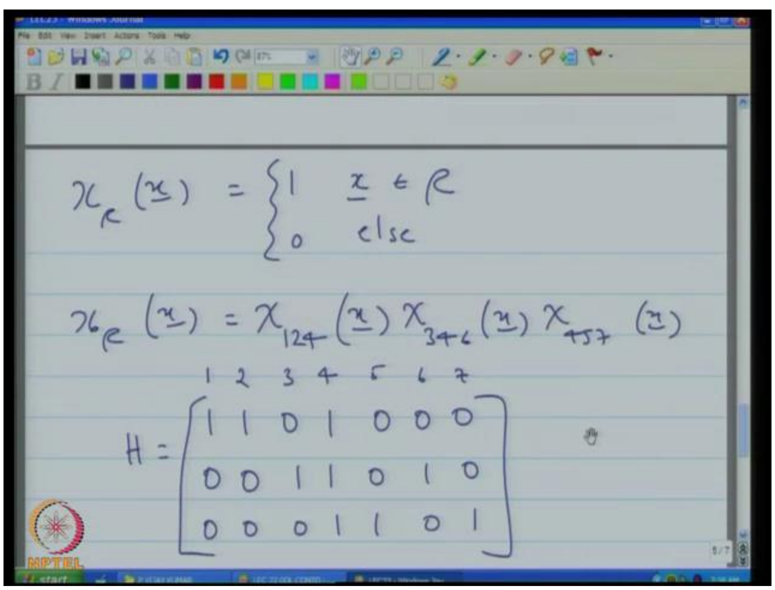
So, the X c of x is 1, if x belongs to the code and 0 else. Now X c of x can be expressed has the product of curve 124 of x chi 346 of x chi 457 of x. Now where did I come up with this? Well, actually come from parity check matrix of this code. So, that is we can bring that down, so we have check parity check matrix in the code. And you can think of see what we are looking for is an indicate function of code. And I am saying this indicated function can be computed by computing the product of three functions and each of this function takes on value zero or one. Now call one to four of X takes on the value one. If X satisfies the first parity check equation may be there X 1plus X 2 plus X 4 equal to zero, and similarly for others. So, let us write that down.

(Refer Slide Time: 21:03)



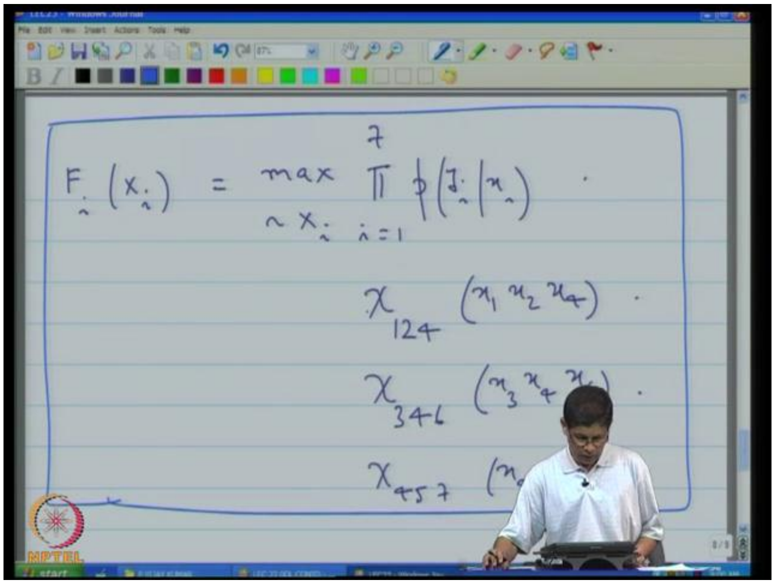
The chi 124 of x equal to chi 124 of x 1 x 2 x 4 is equal to 1; x 1 plus x 2 plus x 4 equal to 0 and 0 else. Shy 346 of x are similarly different… So, this is how, how these are defined. And in all of these is equation here, the addition is actually carried out modulo two.

(Refer Slide Time: 22:42)



But I am not putting down, because I think that is clear. That is particular binary code that just to make thinks little bit more clearly, I let me just number this column here, so 1 2 3 4 5 6 7. So, from this numbering you can see that the first parity check corresponds to the first second and fourth symbols .The second to the third fourth six, and last one to the fourth fifth and seven. Then exactly what have actually put down here. Now with this we can actually rewrite this condition once again.

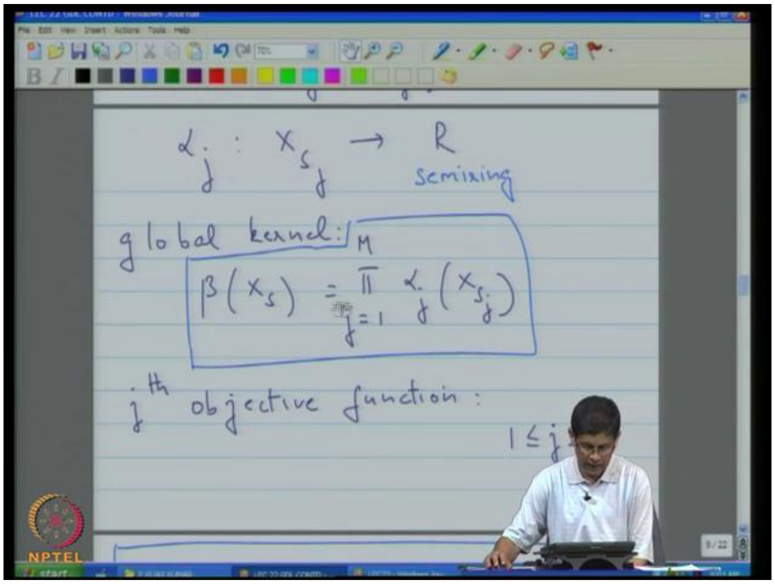
(Refer Slide Time: 23:39)



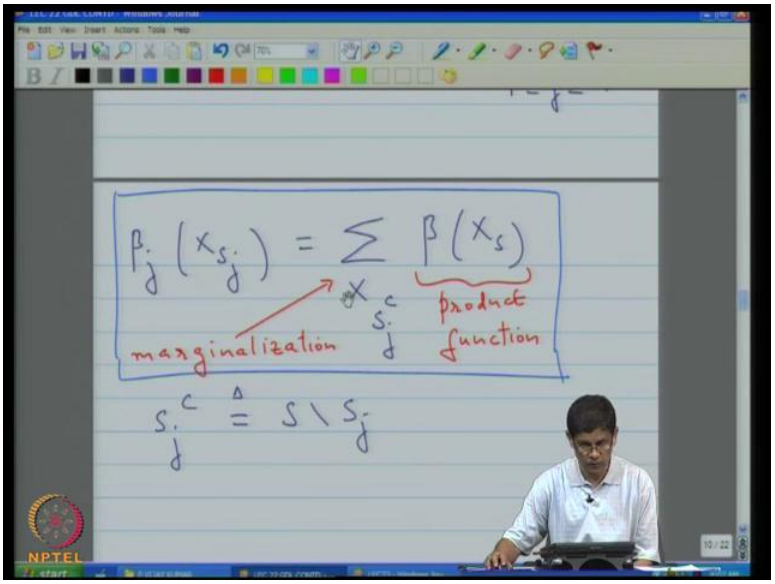
That F i of X i is the max over not Xi, I will explain this in a second. Of P of Y, actually I like to write this differently as well. We will write this is the product i is equal to 1 to 7 P of Y i given x i times. And now, I am going to have chi 124 (x 1 x 2 x 4) times; chi 346 (x 3 x 4 x 6) times; chi 457 (x 4 x 5 x 7). So, we have… So, what is this delta x i stands for? So, what this actually means this that you taking over the max over all components of the vector X other than x i. So, this notation stands for max over X 1 to X i minus 1, X i plus 1 to X 7. So, now if we are taking max over all variables other than X i. So, you only indicate one that is excluded. Then these terms out to be handy notation in our contexts with that the moment you write this, you can (( )) this is an MPF problem, because what you actually have your precisely the local kernels, and each of these local kernels is defined for a local domain. And then you have the global kernel which is the product of all the local kernels, and then you have your working in max product semi ring. So, you have product in the max, and this is vary up carrying out a marginalization which is the summation.

So, parts we can just go back once quickly at the last lecture to look at MPF problem. So, those very minds as self were the setting was. There the MPF problem, we had bunch of we had the universal set and universal domain of variables. We had subset which was called local domains, and then the local domain alphabet is defined here. Each local domain is associated to local kernels and a local kernel takes on value in a semi ring.

(Refer Slide Time: 26:47)

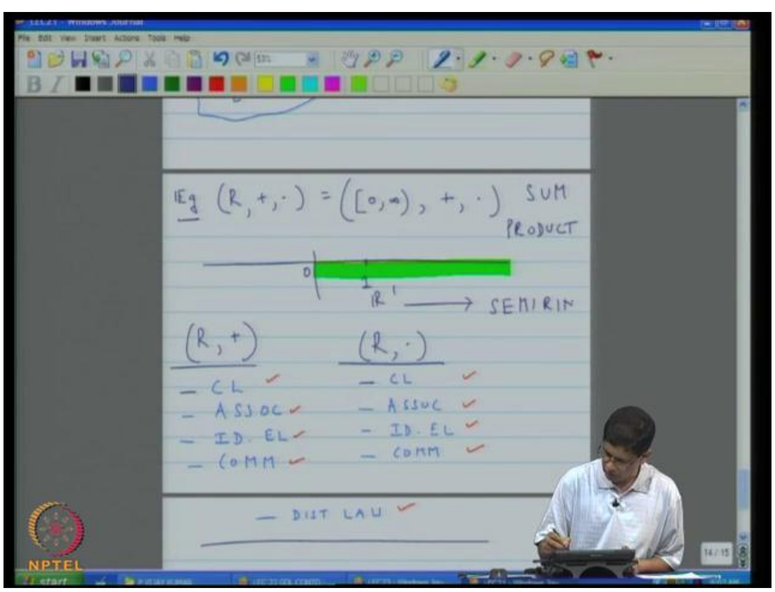


(Refer Slide Time: 26:49)



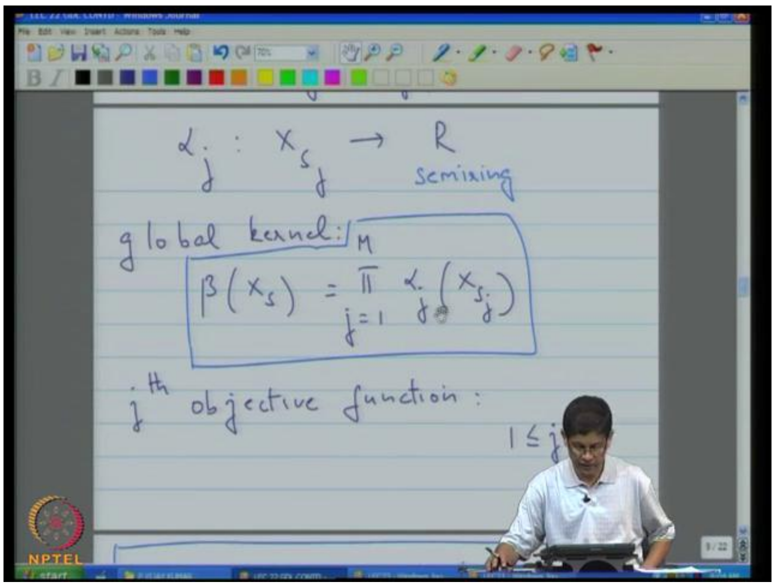
And then you defined a global kernel to be the product of local kernels, and you are objective function has being marginalization of the global kernel with respect to the complement. You get rate of complement of certain variables, if you computing the j eth objective function. That associated to local domain S j, you marginalize where as j complement. Now this brittle product in some here, because as you know there are different types of semi rings in our types, in our lecture twenty one, in lecture twenty one which we see before as now.

(Refer Slide Time: 27:31)



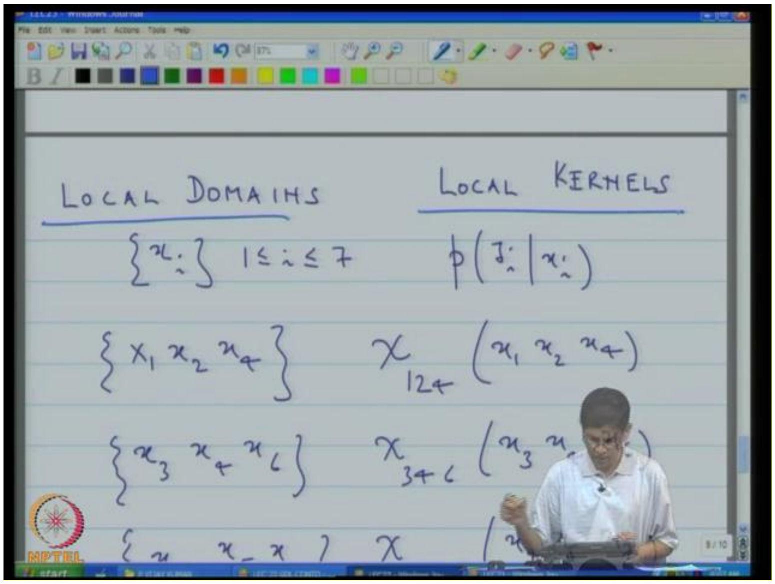
We looked to begin with at the sum of product semi ring, which was which is shown here. This is the range of integer real numbers consider and the operations are plus in that. Let me for emphasis put down, sum product here. So, this is the sum product semi ring. Then in lecture twenty the next lecture. We look at further examples, the further example being the max the max product semi ring where the two operations from max in the product, in that is are current setting. So, when you comment I want to make is that the MPF problem as interested in marginalization of the certain product function.

(Refer Slide Time: 29:00)



We are interested in marginalization of this product function, but when we induced we were unconcisely thinking or sub concisely thinking of sum product semi ring. Because it have to represents something unless you fix something concrete. So, for that reason we actually wrote this summation. But the exact operations are a function of the semi ring. So, in sum product semi ring would have sum over here and the product over here. But in the max product, this would be a product alright. But this would be replaced by summation as it is in our case. In our case we are exactly looking as marginalization of product function. Here is the product being computed here, and here is the marginalization which where summation is now replace by the maximum operator.

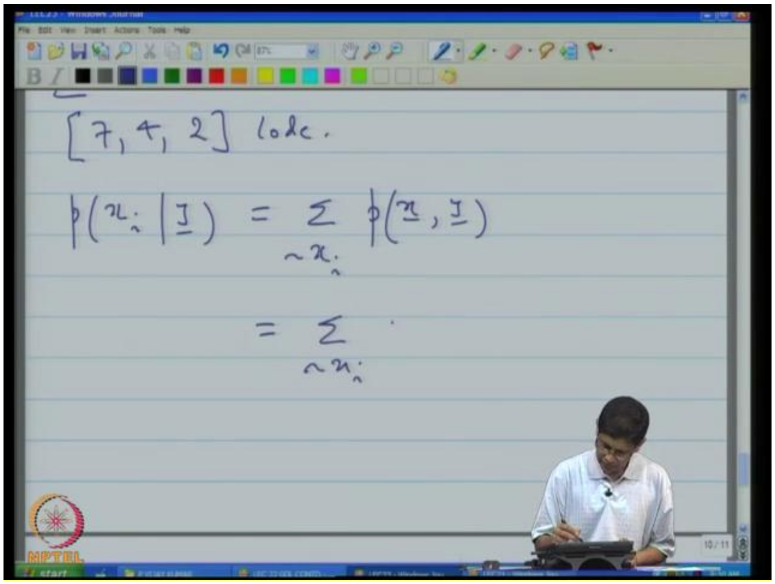
(Refer Slide Time: 30:00)



So, let as put down or local domains or local kernels. So, the local domains are here x 1, x 2, x 4, x 3, x 4, x 6, x 4, x 5, and x 7. And here, you get each of the variables stand alone is local domain, one common which are repeat which have mention earlier. You do not regard the Y i is the variable, because in practice one you imply this Y i stands for receive vector. And then you are carrying out decoding, the receive vector you will operate on specific instance of receive signal which means Y will be a specified real number for that computation.

So, Y is not a variable; it is a fix number. That used in from this term local domain that you extract is just the X i and overall you have an X i on the slide is well. So, local domains therefore are each of this X i where i ranges between 1 and 7. And the corresponding local kernel is P of Y i given X i. Then you have the local domains x 1 x 2 x 4 associated to x chi 124 (x 1 x 2 x 4) and so on. There are local domains and local kernels and we that it is clear that, what we have in fact is an instance of the MPF problem. So, took as while, but variable that to formulated. And it might this stage when you listening to this sum like a lot of notation and lot of work. But when you actually go down to the example and carry out decoding, it find of that proceeds rather smoothly.

(Refer Slide Time: 33:12)



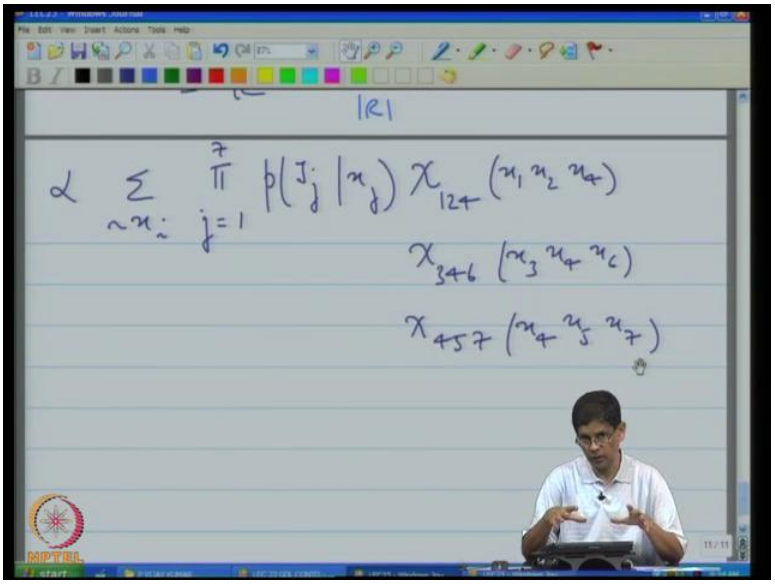
Now, I want to while we are still fresh from working of this code, I want to give a next example also having to do with this code. So, next example will be maximum likelihood code symbol decoding of the 7, 4, 2 code. So, this time what we are interested is once again decoding, this same code. That we have over here. That earlier we have interested in finding out which was the code word that was most likely. It took the code word as a hole. Right now, what we are going to focus on and this is what is commonly now in practice, where going to look at each of the code symbol. And we are going to choose them individually. Earlier we have choosing the entire code word as a whole, but now may not go to do that. We are going to select each code word, code symbol individually.

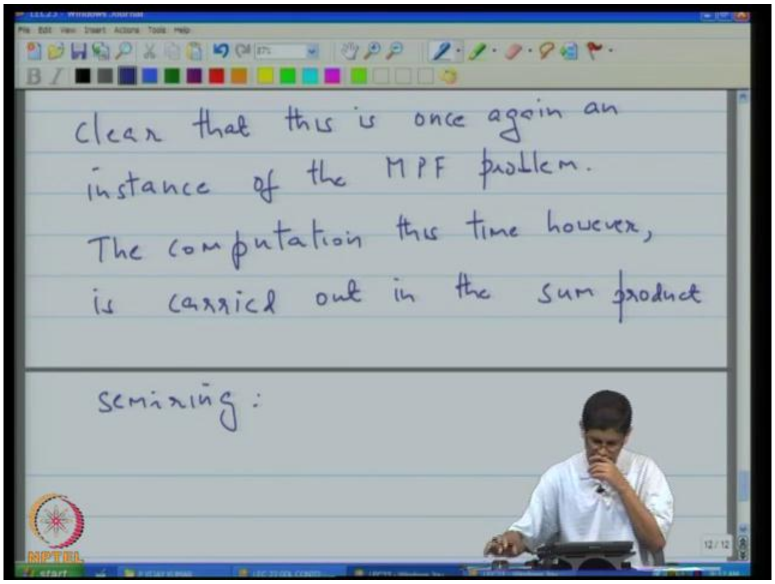
We are going to actually compute where interest in computing P of X i given Y. And we want to compute this for high ranging from one to seven and the way actually compute this is will say that this is the sum over all variables other than X i of P of X comma Y, which is the sum over X i of ... This I guess little bit careful here. We want to evaluate this only for X in the code. So, we write this as not sum over X i of P of X, P of Y given x and x in the code. And let us assume that all code words in the code or chosen equally likely.

This one small note either need here that strictly speaking what I should put here is X given Y. And however and I will explain this in a seconds. So, I am going to introduce the symbol which is stands for proportionate too. So, what you I am meaning that the difference between quantities here. That you are evaluating this quantity is merely that in this case, we have multiply this quantity by P of Y.

But in this computation are interested is in finding out the value of this for X i equal to zero, X i equal to one. So, and then choosing and there looking at which is the larger and accordingly making the decision on X i. If we multiply this entire quantity by some function of Y, that is not going to change things. It will not affect our decision and that is while was freely to do that. Those have written in proportional to, so we come down on here. And now the uses that trick that were used earlier to get rate of this condition, conditioning that you work over the code.

(Refer Slide Time: 37: 45)





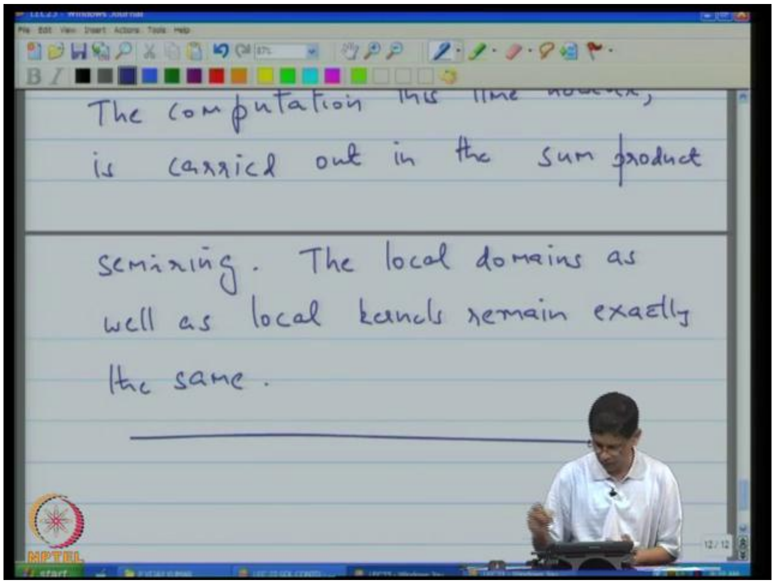
So, you write this as then not sum over X i of the product j equal to 1 to 7 P of Y j given X j chi 124 (x 1 x 2 x 4) chi 346 (x3 x 4 x 5) chi 457 (x 4 x 5 x 7). Now the probability is choosing a code word here.

Assuming that codes are all uniformly chosen with equal likelihood, this quantity is really one upon this size of the code. This quantity is the really one upon the size of the code. But since again this is the constant that will not vector, this is we can ignore this. So, strictly speaking in this computation here of the bottom I should have dividing by one upon the size of the code. That since in this constant I am going to ignore that.

And for that reason, just be accurate. We will compensate, putting second proportional to sign here. Because you are going to ignore precisely, because we are going to ignore this constant, because we know that will affect vector decision. Then this is the problem, this is the computation that will left with for maximum likelihood code symbol decoding. Let copy this observation, the observation is well. This will exactly like the likelihood code word decoding formulation.

So, let us look at that. So, the maximum likelihood code word formulation was like this. So, we have the same quantities that were actually computing the product off. The only difference that here, you have taking the max were as here one is actually taking the summation. So, the difference is between the max and summation over here. So, that means that in this instance we have working in the sum product semi ring. But there is no difference and once again, the local kernels and local domain will remain the same. And excluded once again, this is an instance of the MPF problem. It is clear that this is once again an instance of the MPF problem. The computation this time however, is carried out is carried out in the sum product semi ring.

(Refer Slide Time: 41:40)

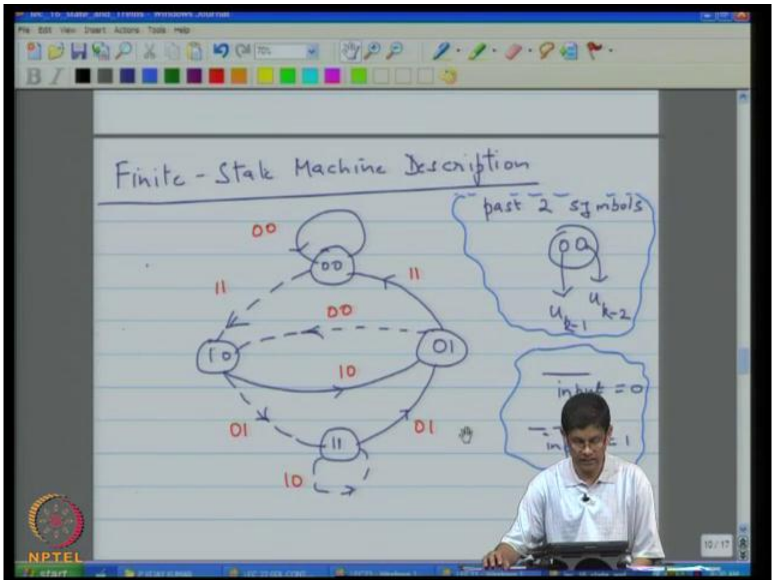


So, what that mean is that for the local domains and local kernels this list remains exactly the same. So, I do not it need to reproduce that. The local domains as well as local kernels exactly the same. We will continue the discussion with further with two other examples having to do with coding theory. The proceeding to example with block codes, some ever will more over. And see that the same formulation also it lies to convolutional codes.

Let us look at maximum likely hood code word decoding of convolutional codes. And what I actually want to do is the convolutional code in this example will be described in terms of states and necessary symbols. So, just give you just to bring this in the perspectives. Let us go back and look at our earlier discussion and convolutional codes. And here we will look at the following finite state machine description of the convolutional encoder.

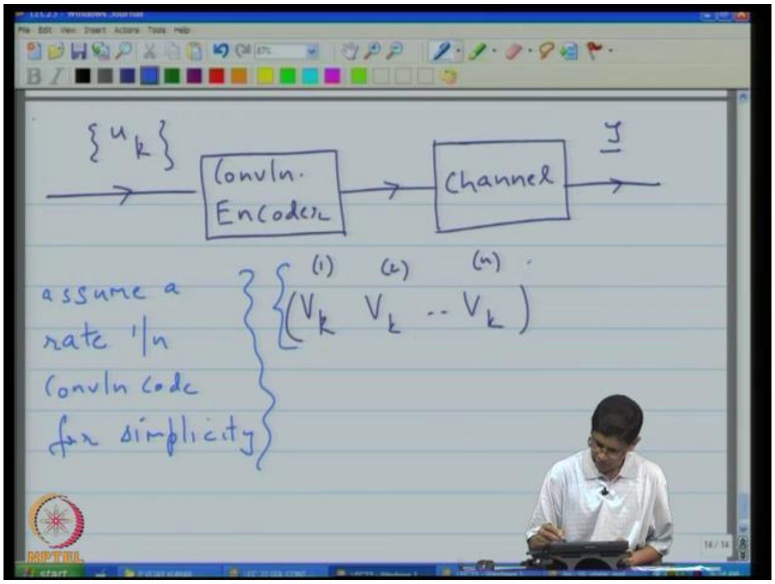
This is same encoder in which we had this. This is these are different state of the same encoder, this is how you would implement in practice. These are the input output recursion in the so called time domain. And this matrix expression requires the input output relationship in the in the D domain which is like transform domain. And this is the finite state machine description associate with their encoder.

(Refer Slide Time: 45:15)



So, all description will be in terms of this finite state machine. And so let us select this page and will copy it over. And I am going to realize, let us keep up with everything. Here is finite state machine and when we just to remind you, these two symbols that we have here other preceding with symbols with most reason symbol being on left side, and the earlier symbol being right side which is clarified here. And the dash dotted in dash lines the solid in the dash line represent on input which is 0 and an input which is actually 1.

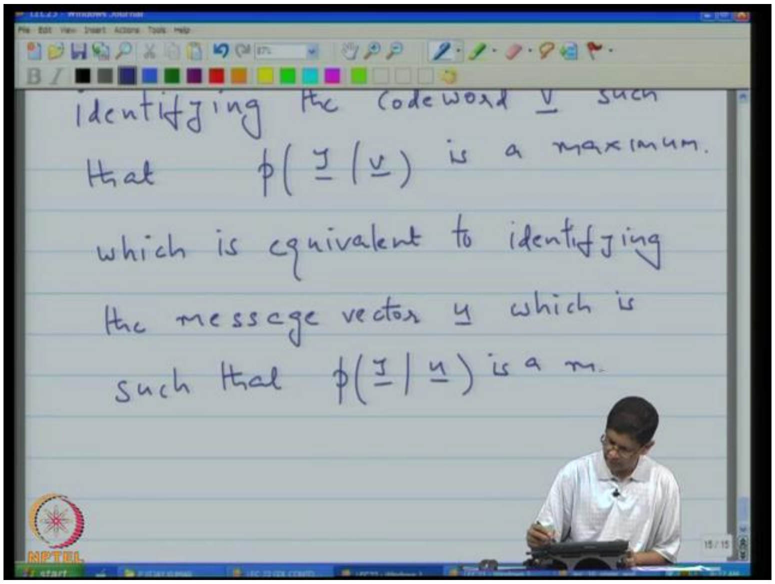
(Refer Slide Time: 46:46)



Let us assume that we are implying the convolutional code word and we have the following block diagram; sum inputs stream of input symbols. These are fed to convolutional code. This is the convolutional encoder and then this is followed by channel, and the outputs we have receive vector Y. And I goal is to carry out maximum likely hood code word decoding of the convolutional code. So, let say that the output of the convolutional code is a stream of output symbol, so let say that this is as was in all case this would be guess, I need little bit most space. So, let me write this here. The output would it be something like v 1 k, v 2 k, v n k. For simplicity we assume a rate 1 by n convolutional code for simplicity. And that is why there is you have stream of input symbol here. For every input symbol here, you have in output symbols.

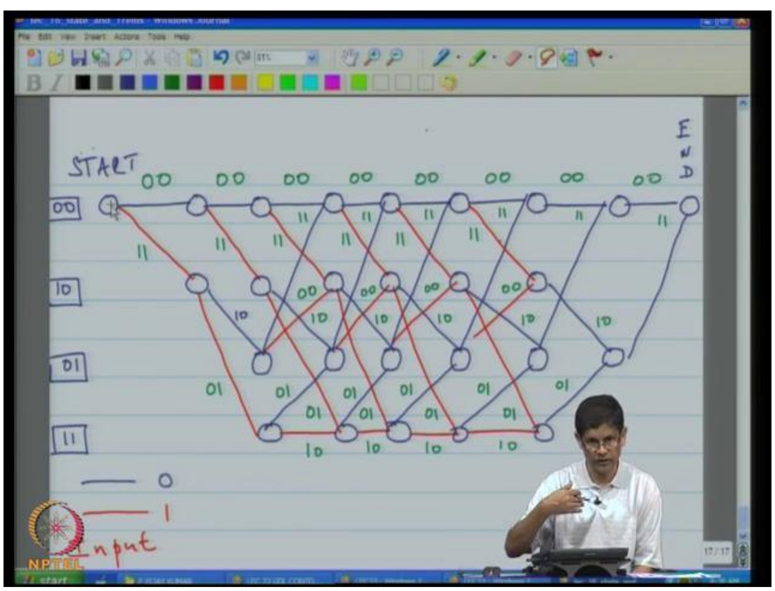
And our goal is actually find out which code word is most likely likelihood code word decoding i e, identifying the code word v such that P of Y given v is a maximum.

(Refer Slide Time: 49:25)



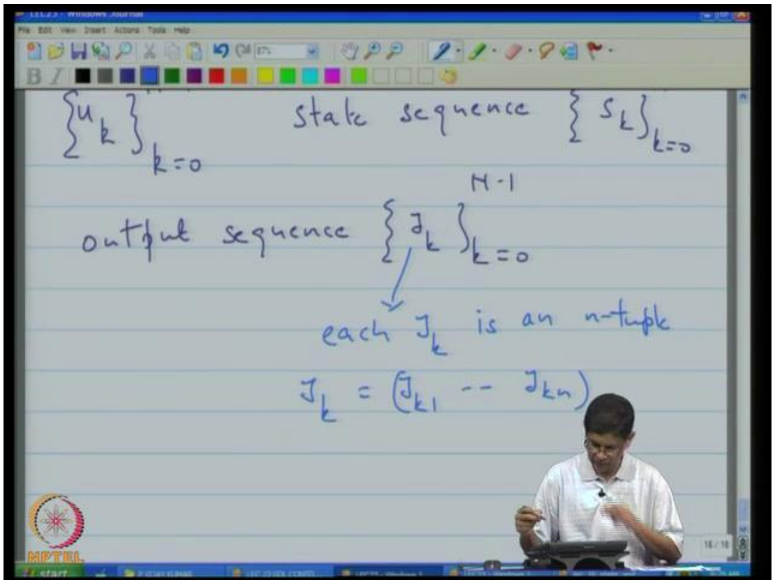
Now this keep in mind that there is a one to one corresponding between message stream and code word stream. One other point which part I should clarify is which easiest described in terms of the Taylors of the convolutional code.

(Refer Slide Time: 51:00)



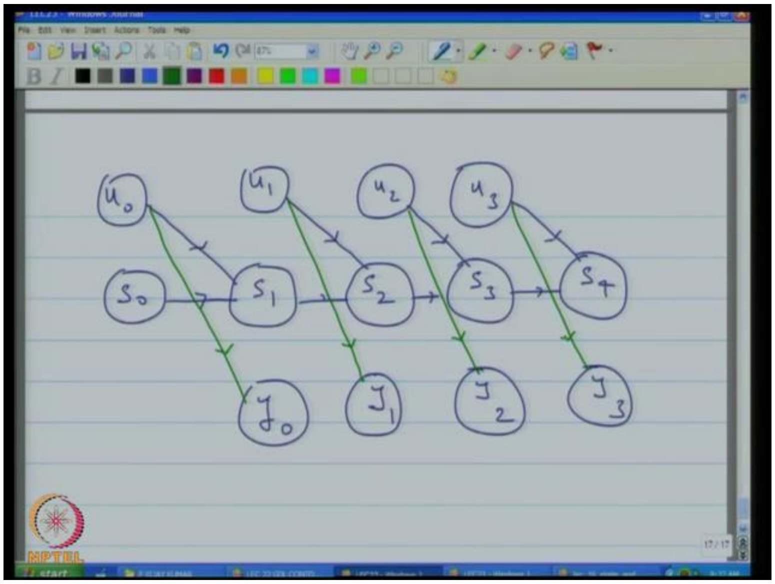
See in this Taylor is you can actually all the code word just being distinct paths in this Taylor. So, each path is associated with a code word, there is also associated in an input stream. So, it is clear that one to one corresponding in an input streams and output streams. So, will actually use that to say which will say this is equivalent which is equivalent to identifying the message … u which is such that P of Y given u is a max. And this will then where goal is it to find the message sequence which is such for the received sequence given that particular message sequence, is a maximum. But we first of all begin with general description the convolutional code.

(Refer Slide Time: 52:44)



So, we will say that I will take particular example. So, in this particular example you have a stream of k is equal 0 to N minus 1 and just as we saw in the state diagram here. Here as each input symbol comes along that the encoder which is finite state machine transmission from state to state. So we will represent state sequence by S k; k is equal to 0 to N. And the output sequence as Y k; k is equal 0 to N minus 1. Now ideality this output will actually be a sequence of vectors. Each Y k is an n tuple. So, we can regard Y k is short form notation for Y k1… Y kn. And can one can described the evaluation of the message and state sequence of the convolutional code by using diagram such as this.

(Refer Slide Time: 54:58)



You have a initial message u 0 and in this example, I am going to set capital N to be… so in the example below, capital N is equal to 3; capital N equal to 4. So, you have a message symbol, and then you also have an associated state sequence, and you have an output sequence. And I am going to draw certain arrows in the state diagram, and I will explain reason for that in the next class, since we are running out of time.

So, I will just complete this diagram and will pick this up in the next class. To summarize, what we did was always took little bit of time, we looked at in the problem of decoding and formulated manage to formulated as an MPF problem, we deduct for two different decoding algorithm for the same block code. Now, we are in the process of doing for this convolutional code. It is hard work, but it is well worth it, as you see when we actually put this formulation to use. We will stop here and thank you.