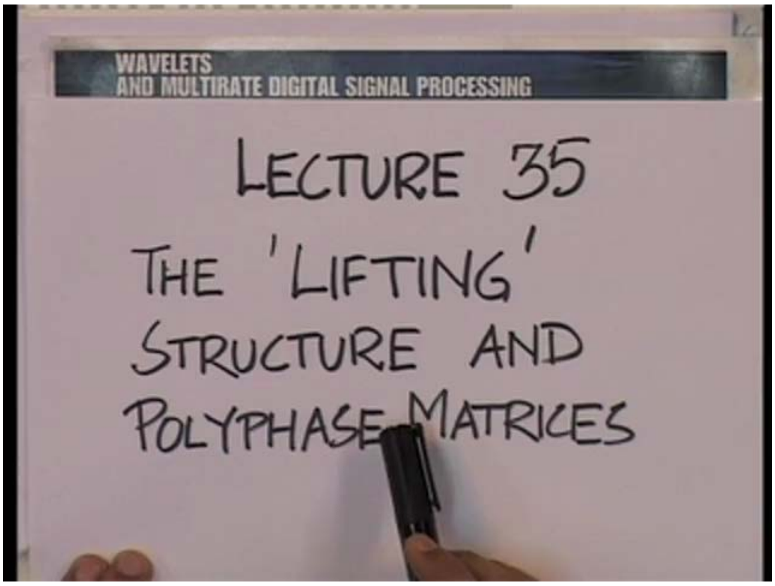
**Advanced Digital Signal Processing –Wavelets and Multirate**

**Prof. V. M. Gadre**
**Department of Electrical Engineering**
**Indian Institute Of Technology, Bombay**

**Lecture - 35**
**The 'Lifting' Structure And**
**Polyphase Matrices**

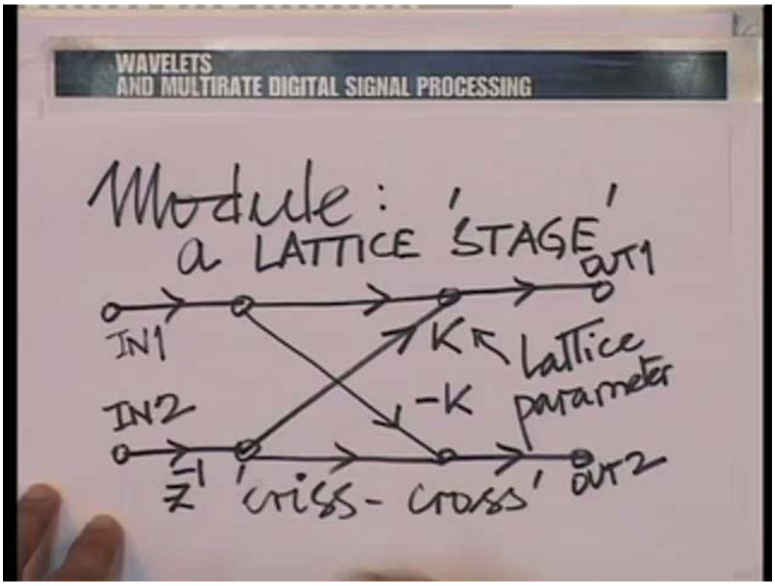
A warm welcome to the thirty fifth lecture on the subjects of wavelets and multirate digital signal processing in the last lecture we had divide up the computationally efficient structure to realize an orthogonal filter bank we called it the lattice structure the word lattice relates to a uniform periodic repetition of a given module or given a modular piece now in that lattice we notice that we had a little bit of complexity the minimum possible where the two input to the lattice interactive with one another to produce the two outputs. In fact, let me put down the structure and let me also put before you the theme of the lecture today the theme of the lecture today is to build one step ahead of the lattice in the lattice there was simultaneous working on the two inputs to produce the two outputs what we are going to do today is to simplify that even further to make it even more modular or even more unitary in terms of operation.

(Refer Slide Time: 02:15)



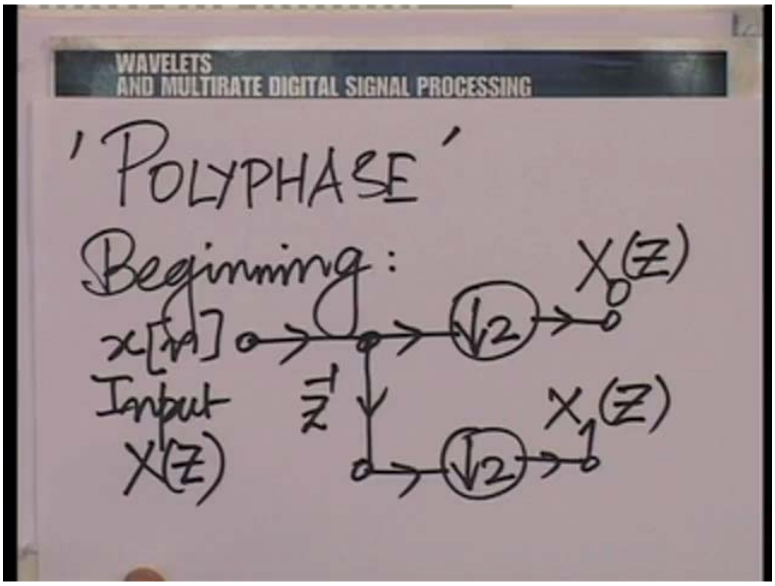

So, we are going to decompose that lattice stage into two sub stages which have even more elementary operations involved and that is going to lead us what we call a lifting structure. So, today we intend to talk about what is called the lifting structure or decomposition of the lattice into even more elementary operations and subsequently we are going to introduce a greater generalization.

(Refer Slide Time: 03:10)



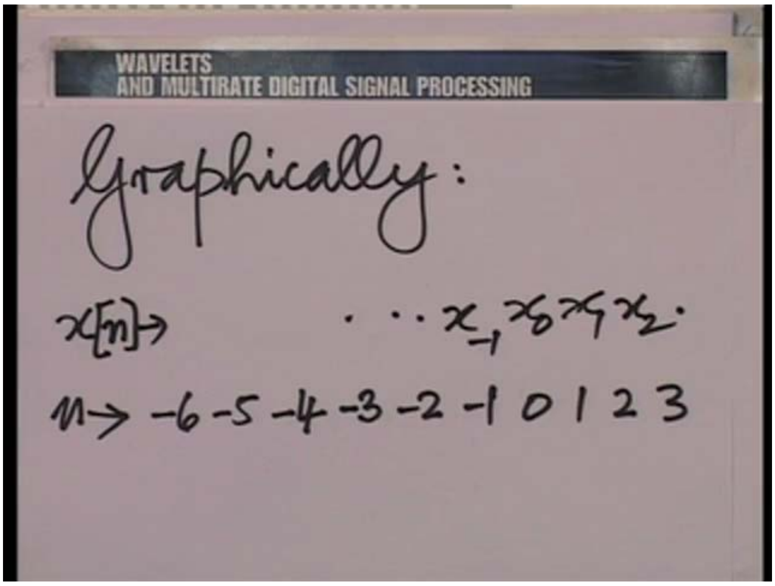

So, we start by a specific modular structure and we are going to a more general formulation of what we expect when we wanted to computationary efficient realizations and that is going to lead us to the idea of poly phase matrices as we mention in the title here now what we done last time was to build a structure that look something like this you know the unit or the module had appearance of this kind and here this was what we called a one LATTICE STAGE and this was called the Lattice Parameter this what this was what distinguish one stage from another the value of k we are also put down a systematic procedure for constructing the lattice and. In fact, we illustrated the calculations involved in constructing that lattice for length four orthogonal filter had left to you to generalize to long the lengths we are given the basic approach and we are shown that generalization was possible now if you look at this lattice stage once again what we notice is that you doing two computation at once you know you are taking a combination of these and produce in this output and you are taking a combination of these introducing this output some sense is a criss crossed involved here now use to think of this in terms of matrices. So, let us call this in input one in one and let us call this in two and let us call this out 1 and let us call this out 2 here let us put down a relation between in and out. So, is this very clear you see what I wish to do is to think of a two cross two matrix that relates the z transforms of the outputs to the z transform of the inputs and this is what we are going to call a polyphase matrix here.

(Refer Slide Time: 06:25)



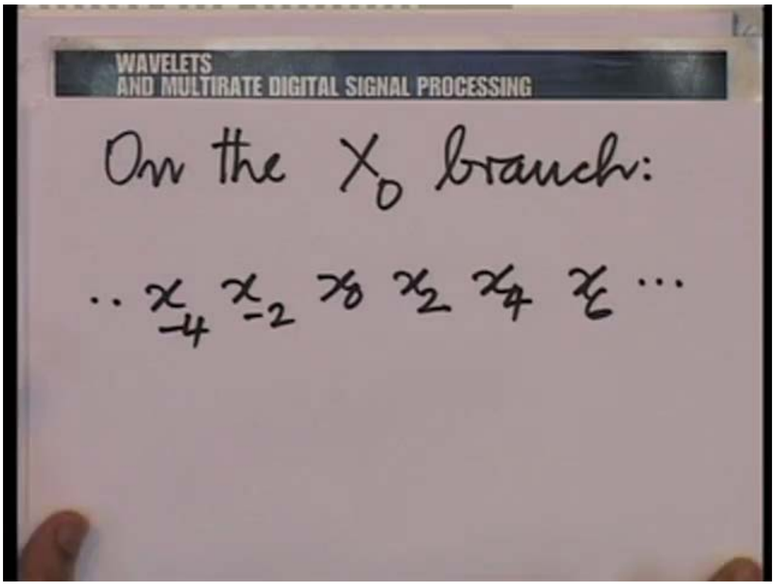

So, you see the idea of polyphase I like to introduce an idea in a little greater depth before in the detail remember even before we put this modules here you know this stage is repeated a different values of k one after the other in cosquite, but at the begin of all these stages is the following operation you have the input a x of n you subject this to one sample delay and then you down sample now if you speak the language of z transforms here. So, if I denote the z transform of the output by capital x of z then one could use the symbols capital x zero of z for the z transforms of sequence appear here and capital x one z for the z transform of the sequence that appears here and we would like to relate capital x zero capital x one and capital x.

(Refer Slide Time: 08:05)



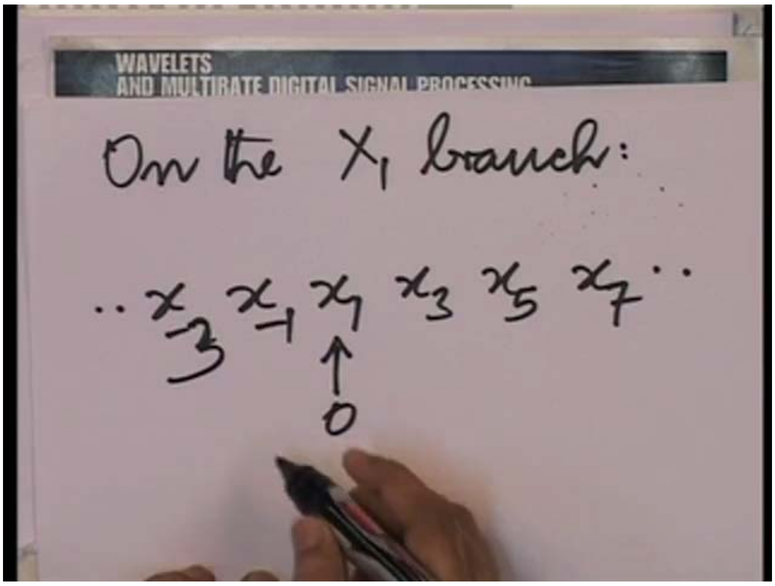

So, let us put down the problem we seek a relation now let us put down graphically the sequences that produce capital x zero and capital x one. So, you know let us write down the indices n of few sample. So, one minus six minus five let us begin with minus six let us say for variety on put in more on the negative side and. So, of course, I have the corresponding samples. So, for example, you have x subzero here x sub one here x sub two there and. So, on z sub minus one and. So, on now what is going to happen on this branch. So, if I look at this branch what I am going to get graphically is x zero followed by x two preceded by x minus two and then x minus four behind x minus six.

(Refer Slide Time: 09:40)



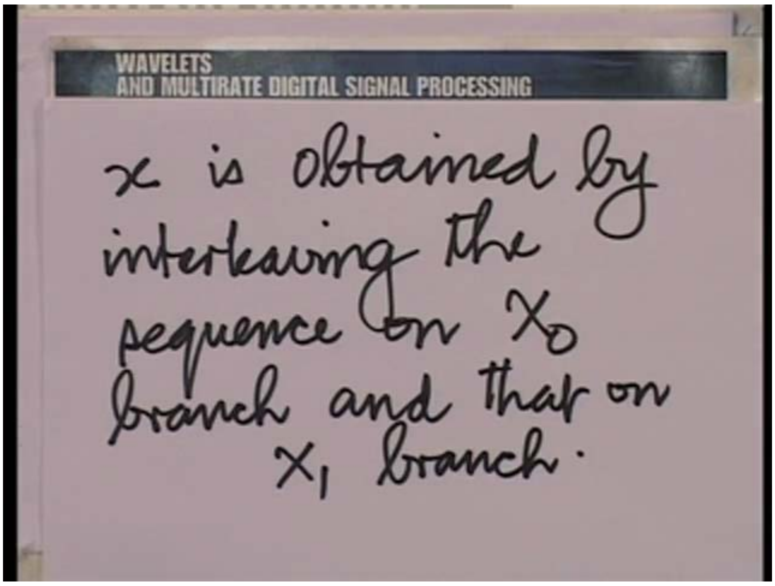
So, so let me right down graphically what I get on the x zero branch I get x zero followed by x two x four x six and. So, on this way x minus two x minus four and.

(Refer Slide Time: 10:07)



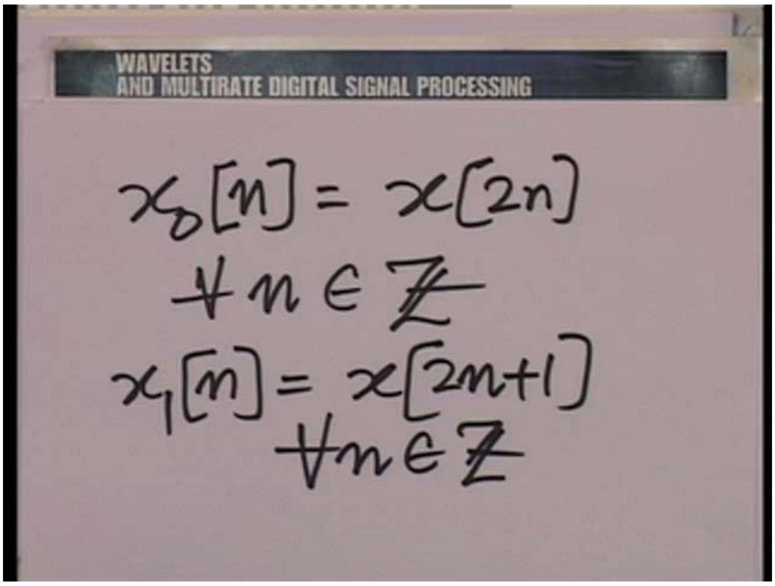

So, on in this direction and x minus one x minus two and. So, on in this direction the only catches radio put the zero now what we do is rather than to call that z transform here as x zero and x one will say their related to the z transforms x zero and x one where we shall give a specific meaning to capital x zero and capital x one. So, for capital x one we shall consider the sequence which is equal to x one at zero subsequently x three x five x seven as shown here and x minus one and x minus three and. So, on behind this should be x minus three there little correction required similarly here on the x branch zero what I am asking for is a sequence that is x zero at the point zero x two x four x six subsequently x minus two x minus four and x minus six and. So, on behind.

(Refer Slide Time: 12:03)



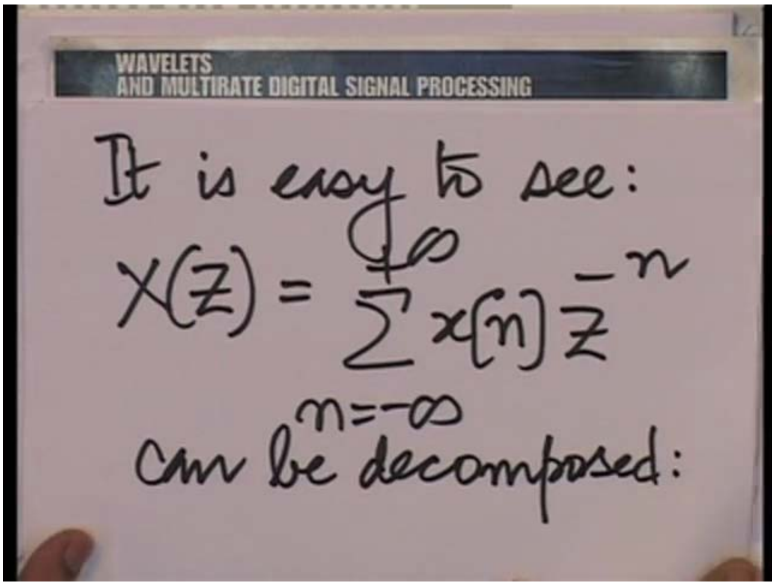

So, what I am saying in a effect is if I consider the sequence x the sequence x is obtained by interleaving the sequence on then zero branch and that on the x one branch. So, what I am saying is essentially if you want the sequence x you take this point from the x zero branch followed by this point from the x one branch followed by this point from the x zero branch then followed by this point from the x one branch and. So, on of course, on easy way of understanding that is to call these sequences let us let us call the sequence x small x zero n and let us call the sequence small x one n.

(Refer Slide Time: 13:25)

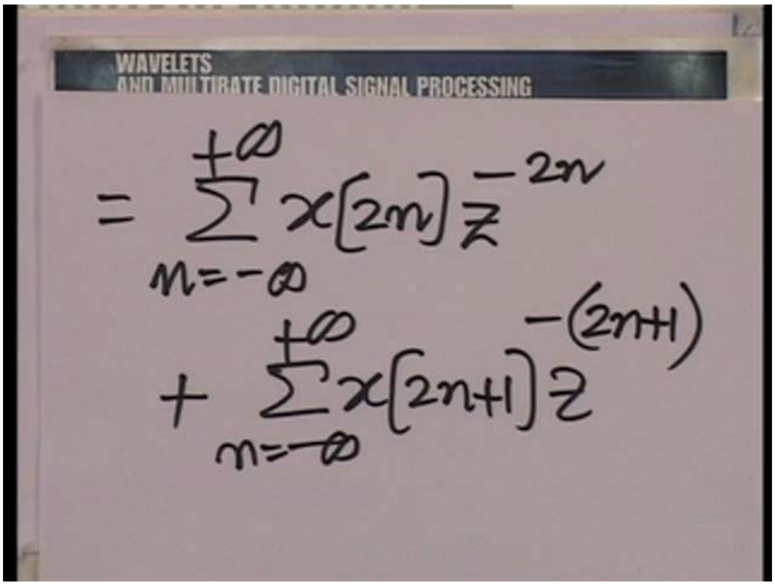


So, in other words we can define x zero n and x one n in terms of the sequence x n x zero n is simply x of two n for all integer n as we got x one n it is x of two n plus one for all integer n and it is very clear that x z

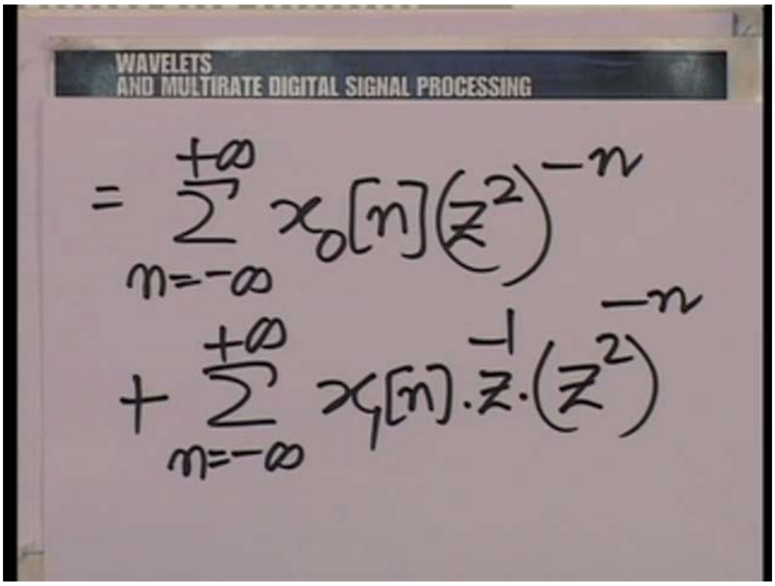
(Refer Slide Time: 13:50)




(Refer Slide Time: 14:40)

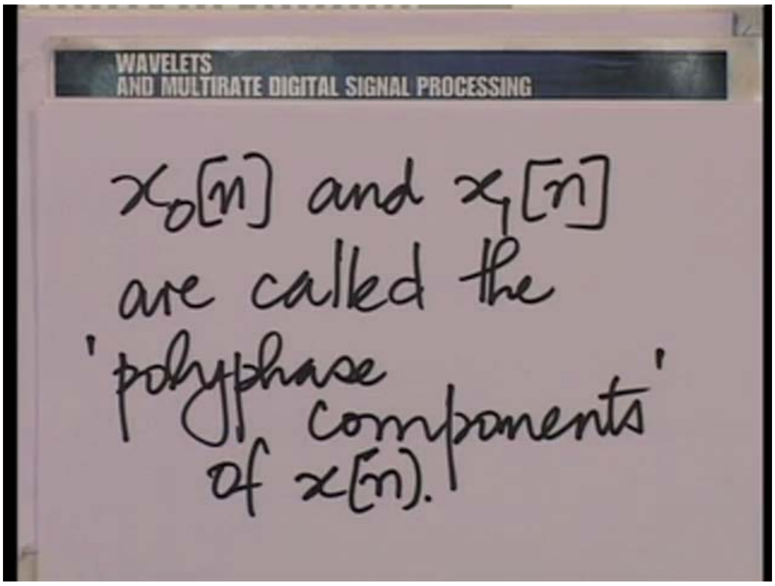


(Refer Slide Time: 15:30)



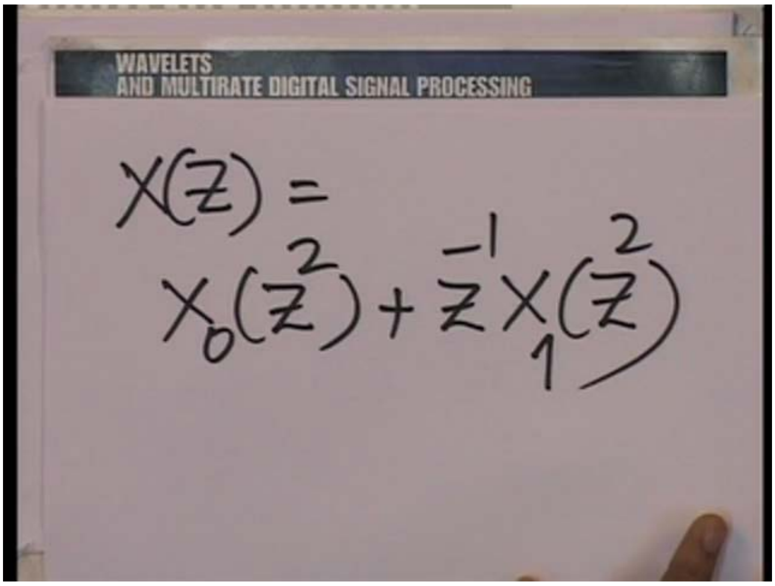

x of z which is summation n going from minus to plus infinity x n z race to the minus n can be decompose into two summations x two n z race the power minus two n plus summation n going from minus to plus infinity x two n plus one z race the power minus two n plus one by definition and very clearly since x two n is essentially x zero of n and x two n plus one is essentially x one of n let me put back the definition before you x two n is x zero of n and x two n plus one is x one of n and therefore, this becomes x zero n into z squared race to the minus n plus summation n going from minus to plus infinity x one n z inverse and then z squared race to the minus n all the type done here is to rewrite this a little bit rewrite this a little bit note that this is essentially x zero of n and this is essentially x one of n to get here now here I am trying to bring out a new idea the idea of what is called a polyphase components

(Refer Slide Time: 16:50)



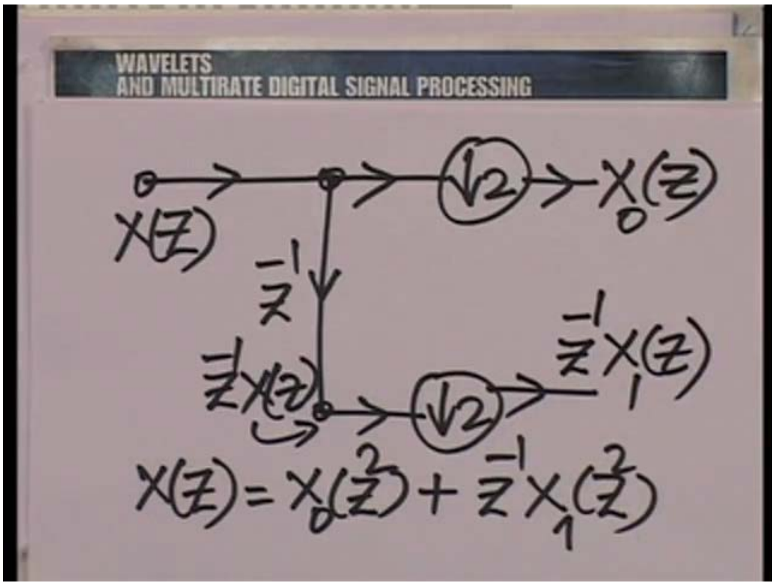

I shall now define this components x zero n and x one n that we speak about here are called the polyphase components of x now what the polyphase comes from the idea of a number of phases is being operated in pram. So, you know if you think about it in the sense of a machine suppose you had the sequence of x coming sample after sample and you wanted to extract the sequence x zero and the sequence x one think of two branches two streams going out of x one is x x comes in here x zero goes on one branch at x one goes on the other it has if you had a switch a polyphase switch you know this ideas of polyphase come from the idea of switching from one phase to the other. So, you switch on to one phase from one sample to the other phase for the other sample again to the first phase for the next sample and then to the second phase for the other sample that is way the idea of polyphase component comes from a switching mechanism between phases that essentially what is happening in constructing x zero and x one.

(Refer Slide Time: 18:45)



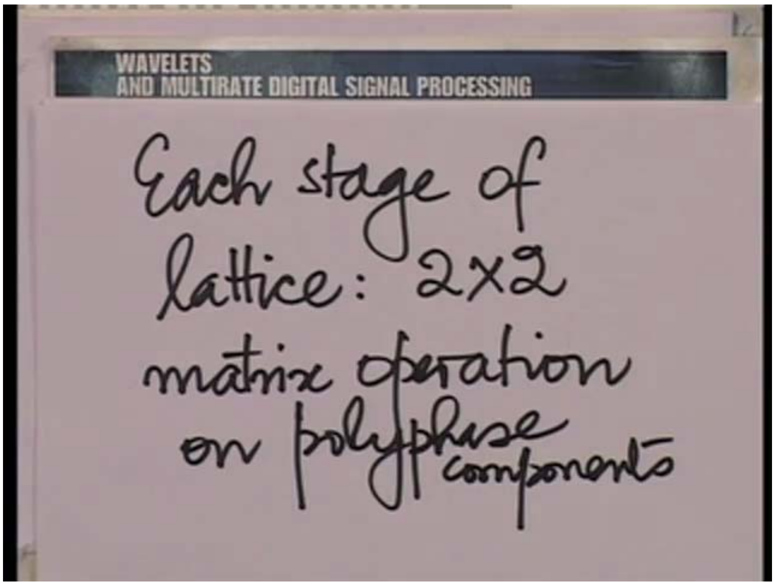

So, it is very clear that the z transform of the polyphase component relates to the z transform of the input in the following way capital x of z if just shown is capital x zero of z squared plus z inverse capital x one of z squared. So, essentially is the relation between the poly z transform of the polyphase component and the z transform of the (( )) now what we are doing in one's stage or one module of the lattice is to operate on the polyphase components so. In fact, you could think of the whole of the analysis filter bank of that matter the whole of the synthesis filter bank also as on operation essentially on polyphase components instead of thinking of it as on operation on the original sequence it could be thought of as an operation on two cross two sequence.

(Refer Slide Time: 19:55)



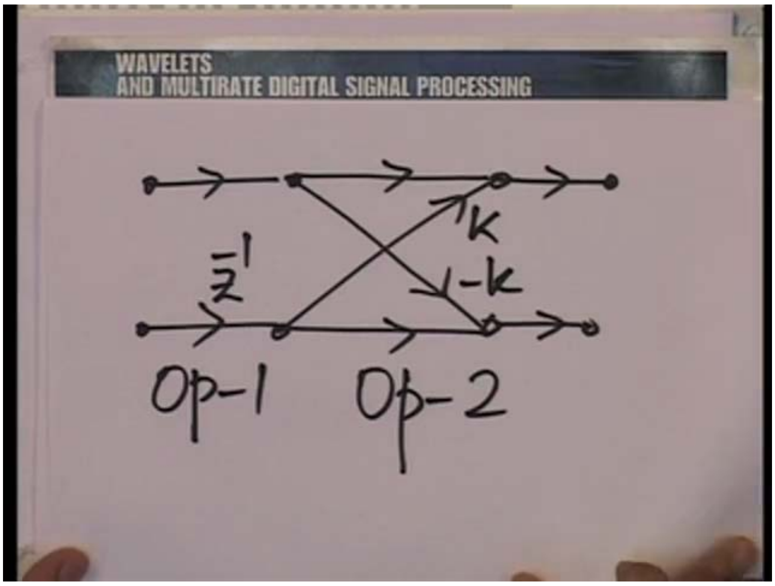

So, if you look at it carefully when we do this in the beginning and we wish to express what comes out here and here in the language of z transforms noting that x of z is x zero z squared plus z inverse x one z squared what comes out on this branch is essentially x zero of z now what comes out this point is z inverse of x z and therefore, you have z inverse factor multiplying this which puts it into odd location. So, when we down sample those are all destroyed, but this becomes the z race the power minus two times x one z squared. So, so when we down sample what we get here is essentially z inverse x one z that is interesting. So, what we are doing essentially is to operate on x zero z and x one z. So, you know the first stage of the lattice of that any stage is essentially an operation on this polyphase components.

(Refer Slide Time: 21:33)

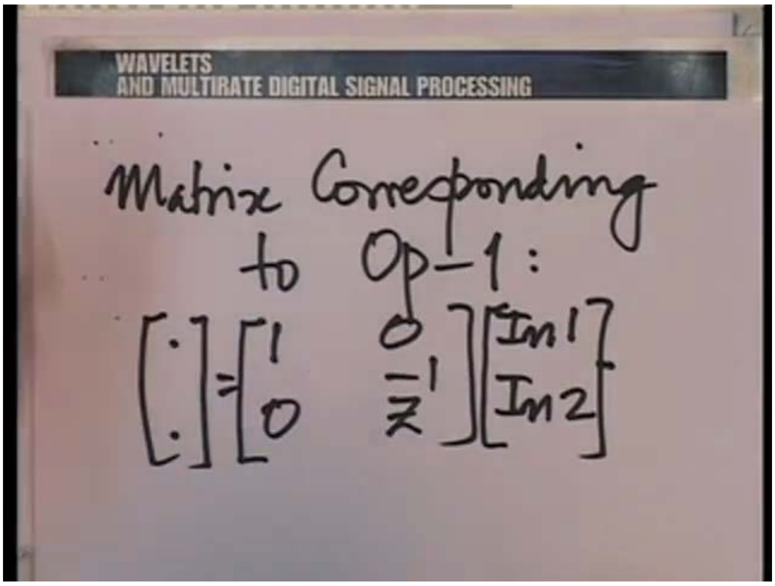



So, we could think of each stage of lattice as essentially a matrix the two cross two matrix operation on the polyphase components let us try and speak this language of matrices for example, when we simply delay the lower branch with the z inverse and leave the upper branch and change the corresponding matrix is essentially a diagonal matrix.

(Refer Slide Time: 22:22)

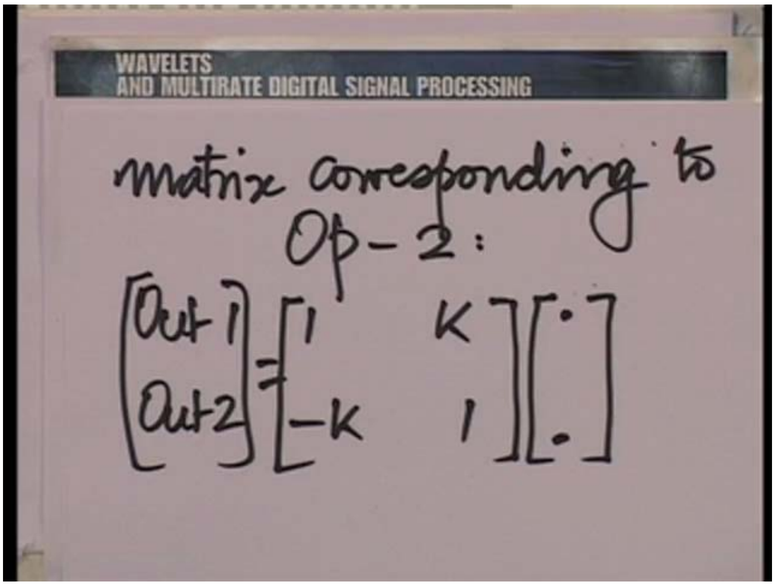


(Refer Slide Time: 22:58)



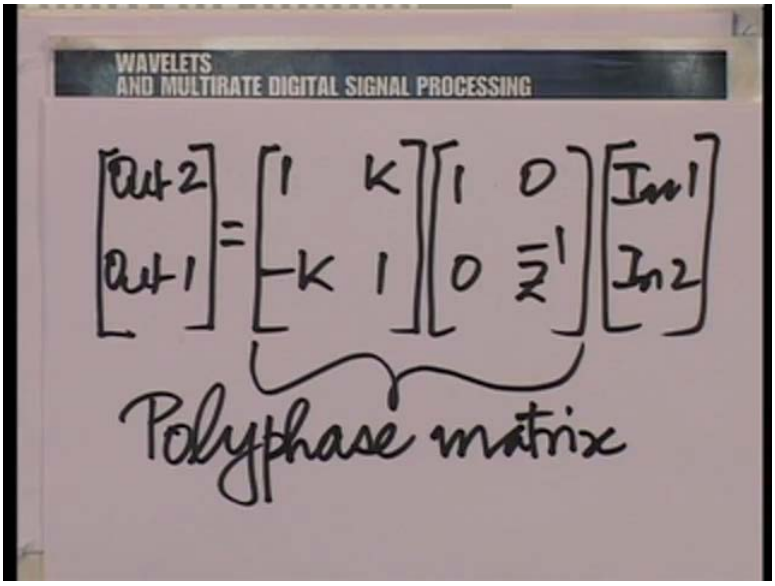

So, what I am saying is this stage here I would not kid drawing the. So, criss cross stage here is one stage and this stage is another where we have z inverse here. So, we have two two sub stages again. So, you can call it operation one and operation two for the k n minus k there the matrix corresponding to the operation one is essentially keep the upper branch as it is and multiply the lower branch by z inverse simple enough. So, what I am saying is you know if you call this in and out in one in two and out one and out two not quite there, but intermediate. So, this is in one and in two and what I am producing at this point is essentially these two outputs here the matrix is this correspondingly the matrix corresponding to operation two can equally easily be return down.

(Refer Slide Time: 24:10)



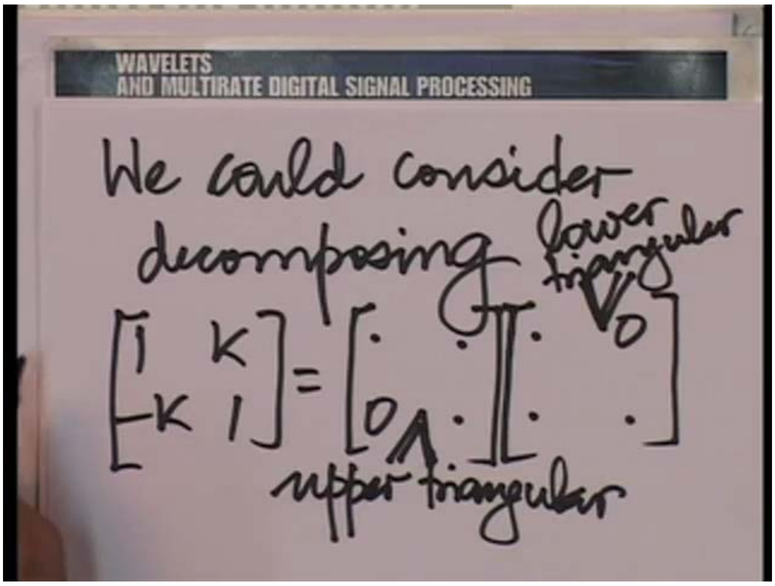

So, if we taken the operation one and on them to produce the upper branch if we taken one times this and k times the lower branch for the lower branch it is minus k times the upper and one times the lower and this gives us out one and out two look back again at the drawing to convince yourself. So, will you take this two and introduce out one and out two, but we are saying is that out one is produce by one times this plus k times this plus this one times this that is precisely what is reflected in the matrix here one times this k times this minus k times this plus one times this.

(Refer Slide Time: 25:40)



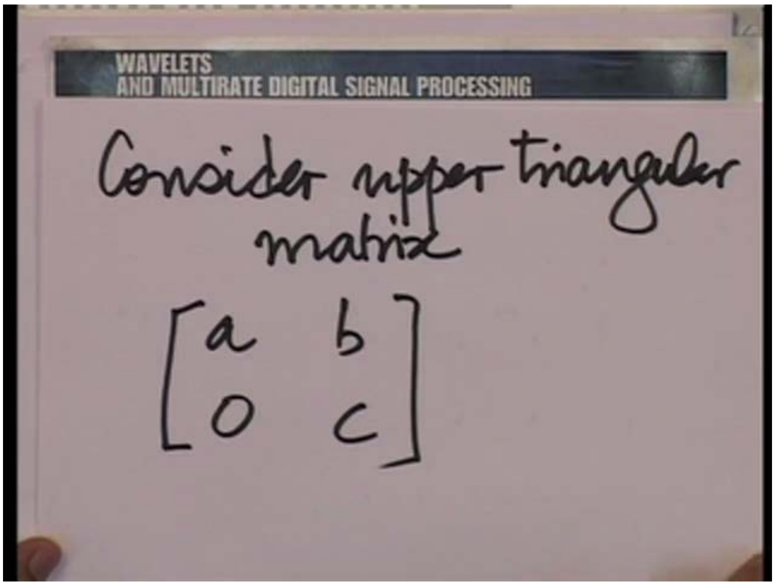
So, compose it operator between in one in two out one and out two is as follows out two out one is one k minus k one one zero zero z inverse in one and in two now this entire operator here a matrix operator acting on the polyphase components is called a polyphase operator a polyphase matrix so. In fact, each lattice stage has a polyphase matrix corresponding to (( )) and when we cascade we lattice stages we are placing these polyphase matrices in cascade two we are essentially multiplying the matrices one after the other that is the interpretation in the language of matrices when we look at it from matrix perspective thinks become much easier to understand we also see what is elementary about these operation these matrices are indeed very simple matrix for example, one of them is a diagonal matrix the other matrix is essentially you know it is essentially with only one parameter two of the entries are one; that means, there is no multiplication involved the other two entries are identical up to a sign sign change these are very elementary simple matrices and what we have done in the process of constructing the lattice structure is to break down the entire low pass and high pass filter on the analysis side into small matrix operator of this kind now what we are going to do now is to ask whether we can even further simplify and break up these matrix operators. So, let us look back at the operator here is just two simple to break down any further there is little that we can do to simplify this, but we could possibly consider simplifying this this is still in some sense of full matrix a matrix of full rank you know all the entries are non-zero and the entries none of the entries are trivial in the well these two entries are trivial, but you know a full amount of computation to do here we can decompose this matrix into a product of two even simpler matrices.

(Refer Slide Time: 28:45)

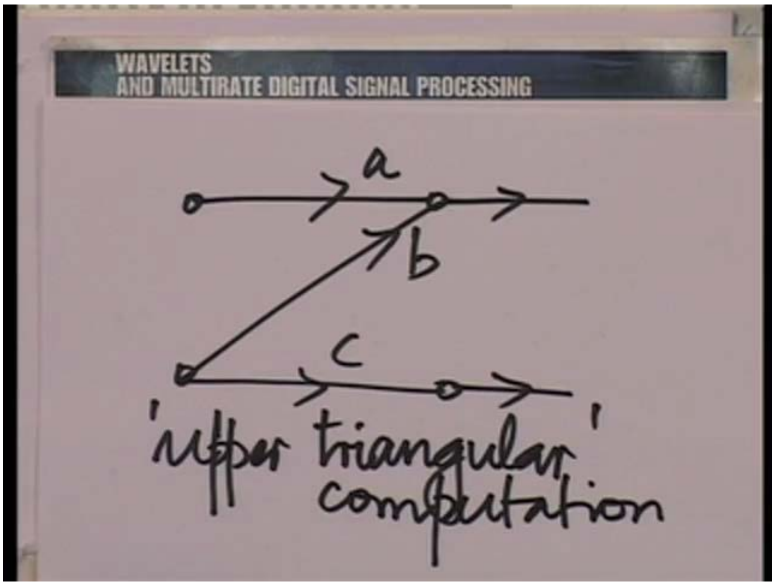



So, we could consider decomposing one k minus k one into two sub matrices like this one matrix which has a zero here and non-zero entries there and one matrix which has non zero entries here and a zero entry there this is called a upper triangular matrix and this is called a lower triangular matrix. So, what we are asking for is essentially a decomposition of this rather simple two cross two matrix into even simpler two cross two matrices one upper triangular and one lower triangular what is it mean in terms of a computation when we have let us say for example, an upper triangular matrix as is here. So, let us let us put numbers in this matrix or at least symbol if not numbers.

(Refer Slide Time: 30:18)

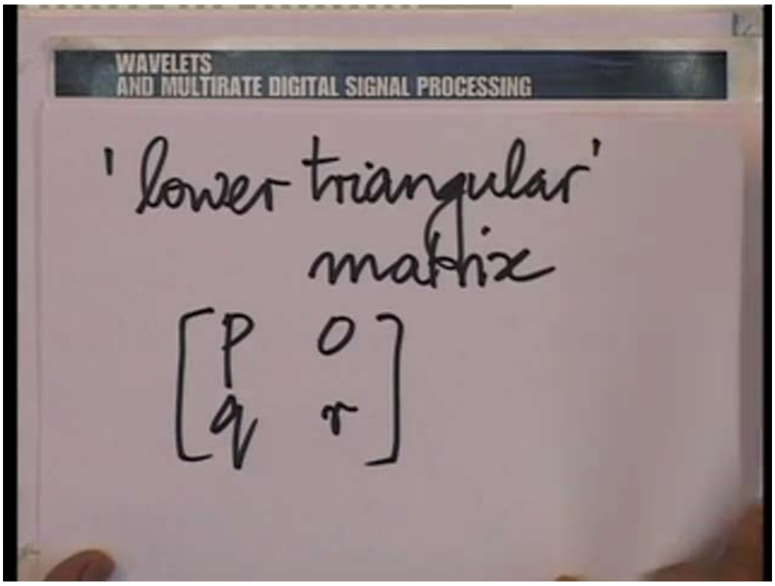


(Refer Slide Time: 30:40)

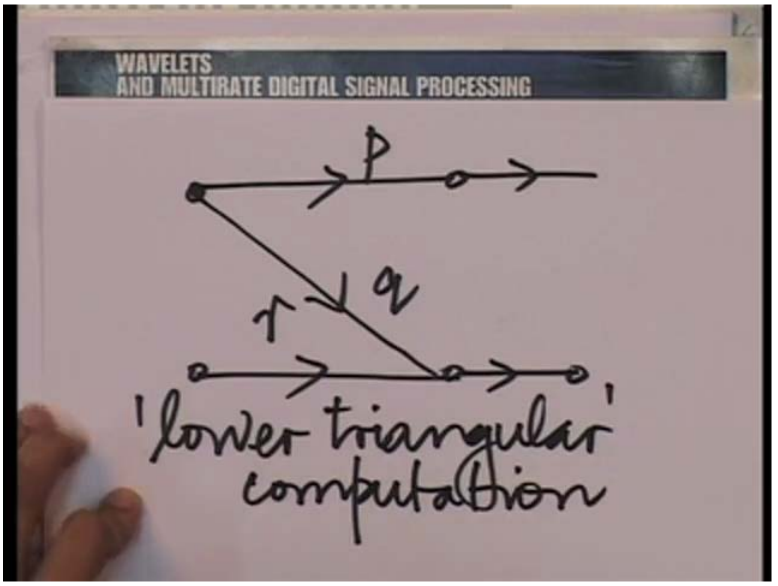



So, consider the upper triangular matrix let it have the form a b c as zero here what is the computation meant the computation meant is the following we are saying a upper branch is eight times this plus b times this the lower branch is just c times this. So, this is what is essentially a. So, called upper triangular computation and similarly we can put down a lower triangular computation.

(Refer Slide Time: 31:30)

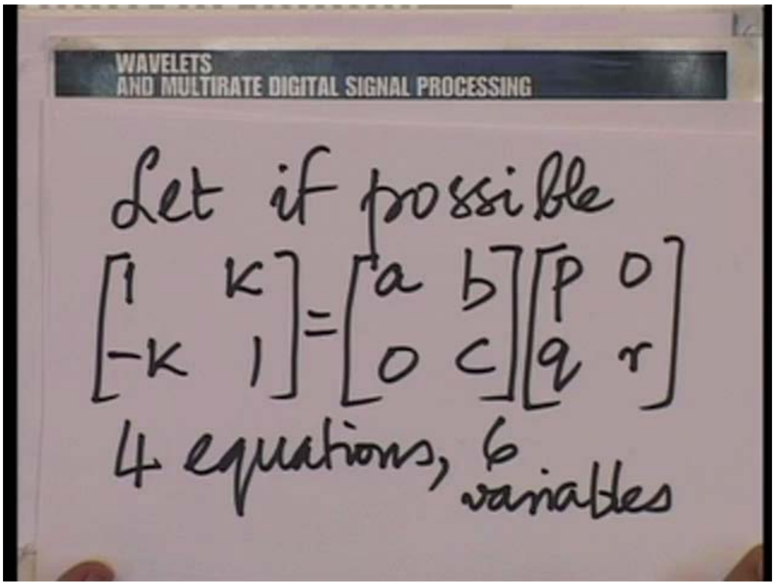


(Refer Slide Time: 31:45)



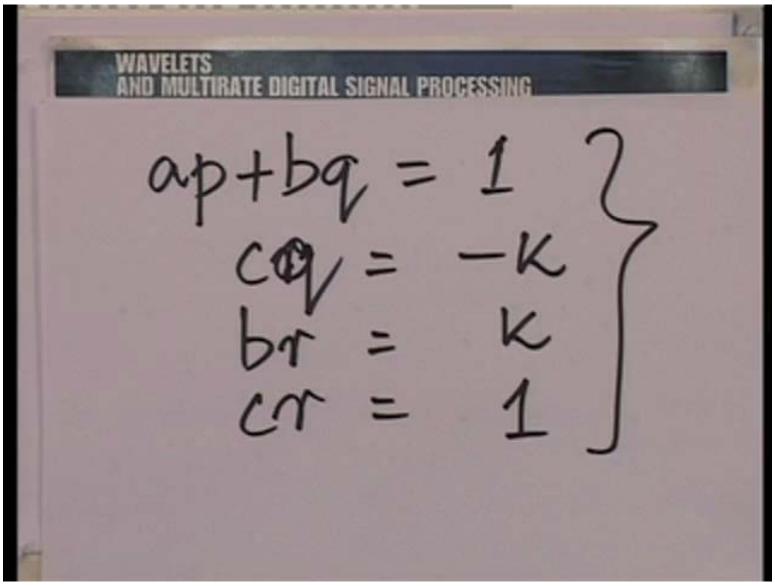

So, typically a lower triangular matrix will look like this with other with associated computation looking as follows now what is interesting about both the upper and lower triangular computation here is that we have further simplified what was essentially a criss cross operation before this. So, it is the most elementary think that you can do if you want to do something that can help you progress it essentially means make a linear combination or make an operation are only one of the branches combine it with the other. So, at one time we do only one combination we do not let branches interact both with each other at once this is even more elementary then the lattice and what we want to investigate is whether it is possible to decompose each lattice stage into these upper and lower triangular forms. So, let us. In fact, assume that is possible and then find a set of values for it.

(Refer Slide Time: 33:30)



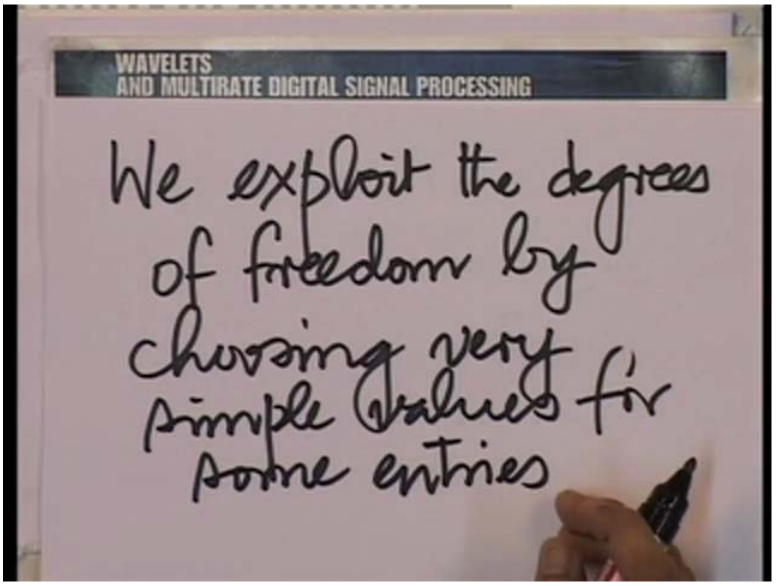

So, let if possible this elementary part one k minus k and one be equal to a b c and p q r written this way first a lower triangular operation followed by an upper triangular operation of course, one could possibly conceive a reversal of roll here, but that is a different issue so. In fact, now we have four equations and six variables; obviously, this has some decrease of freedom in the solution we are exploit those decrease of freedom in a minute, but let us write down the equations at we get from here express it. So, how would we get the equations we would get them by comparing term by terms for example, one is a times p plus b times q minus k is essentially what you get by a dot product of this with this. So, three times r and. So, on.

(Refer Slide Time: 34:55)



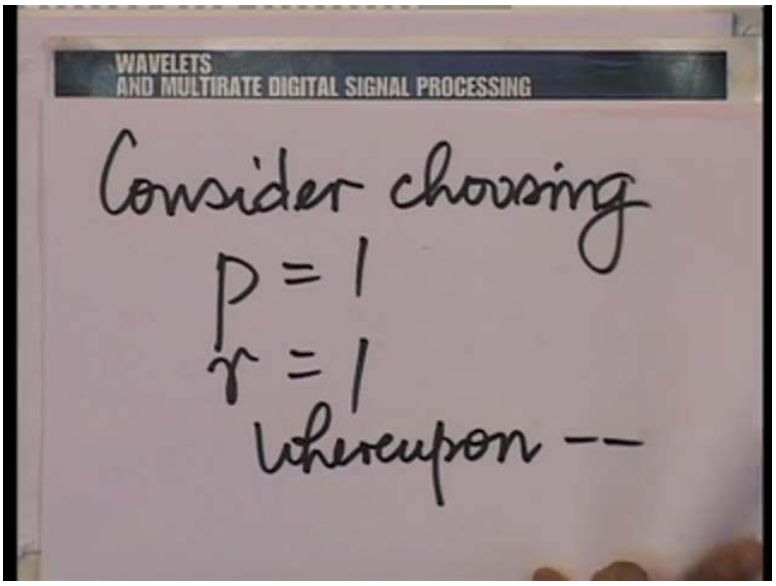

So, let me write down the four equations at we get a p plus b q is equal to one c r is minus k b r is k and finally, well the last one well just a minute I think there is a slight mistake minus k that is right. So, that is c r c q is minus k. So, you have this into this. So, zero times p plus c times q. So, c q is minus k b r is k and c r is one is the four equations and as exceptive we have two degrees of freedom here we can only solve for four variable. So, we are free to choose two variables in a way that deem appropriate now if you look at this operation the first operation here the lower triangular operation what we would like to see is whether we can avoid these multiplications here. So, you know we would like to make this a lower triangular operation as simple as we can if I go back to the actual computation suppose we could ensure that one of the at least some of these are just one you know these multipliers here if you could make them one and get a solution for the other what it means is that for example, if p could be made one wages passing it as it is if r could be made one wages passing this as it is. So, we exploit it in these equation.

(Refer Slide Time: 37:13)

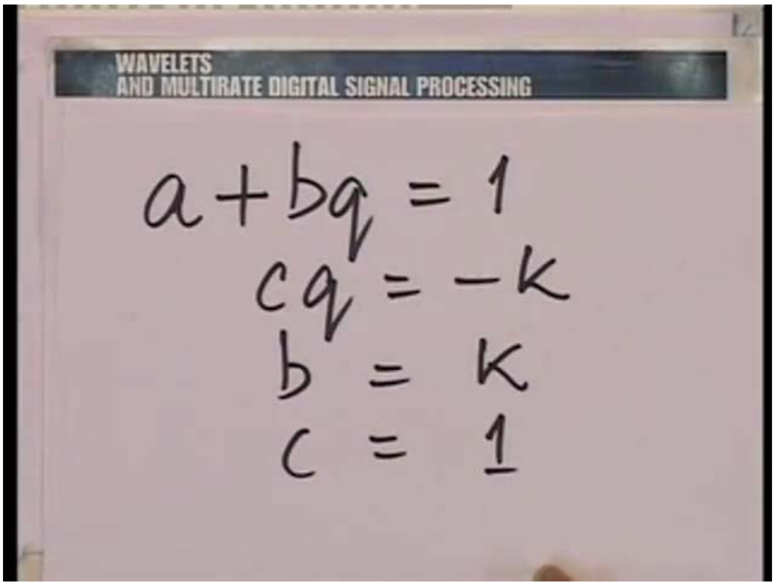



We exploit its degrees of freedom by choosing very simple values for some variable.

(Refer Slide Time: 37:40)

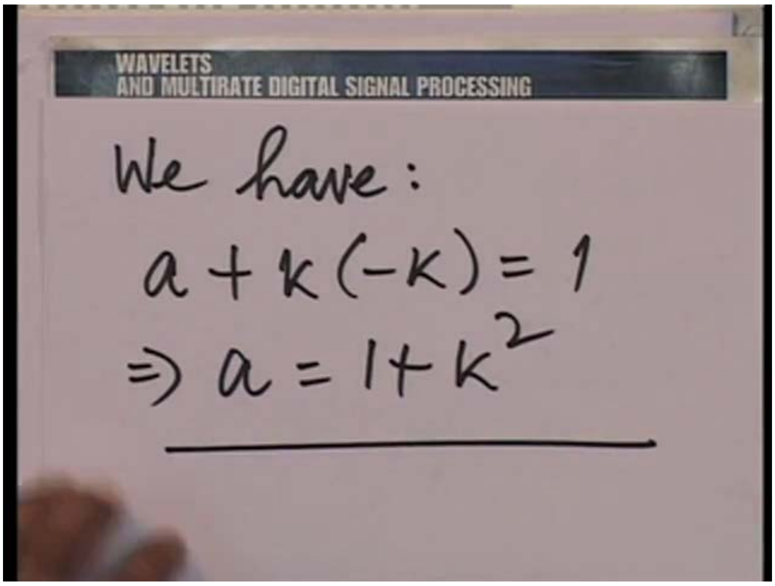


(Refer Slide Time: 38:15)

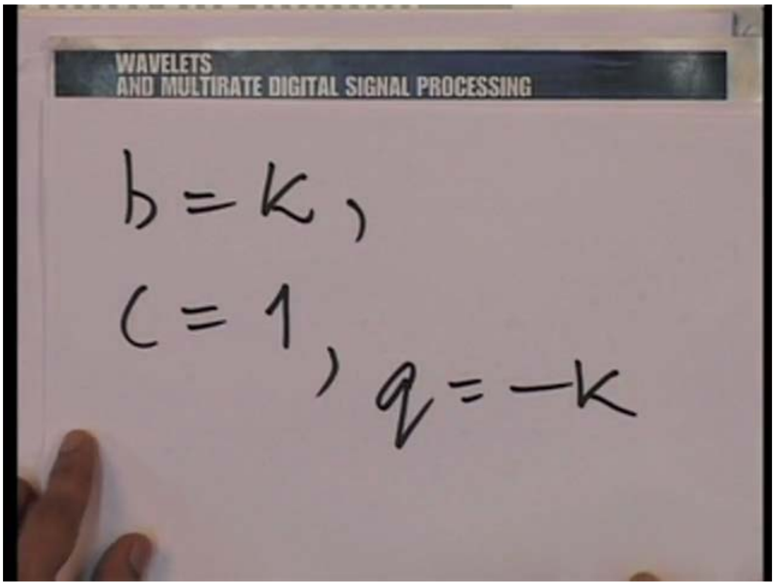



So, for example, let consider choosing p equal to one and r equal to one in this lower triangular form. So, make the diagonal entries one here where upon here a plus b q is one c q is minus k b is k and c is one if we substitute back in these equations remember p and r have been made one and this is very easy to solve now c is one. So, of course, q is minus k when q is minus k you know what to do here and b is k. So, all which we need to do is to essentially collapse all these equations into this.

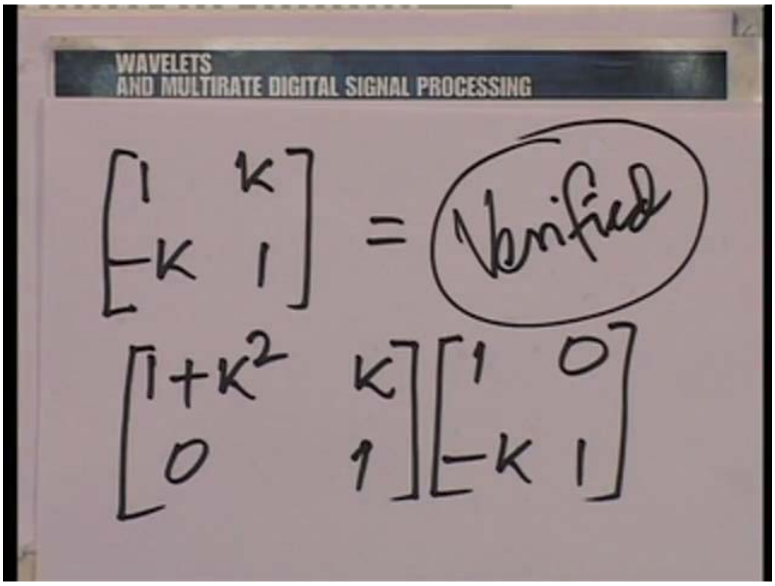
(Refer Slide Time: 39:05)



(Refer Slide Time: 39:25)

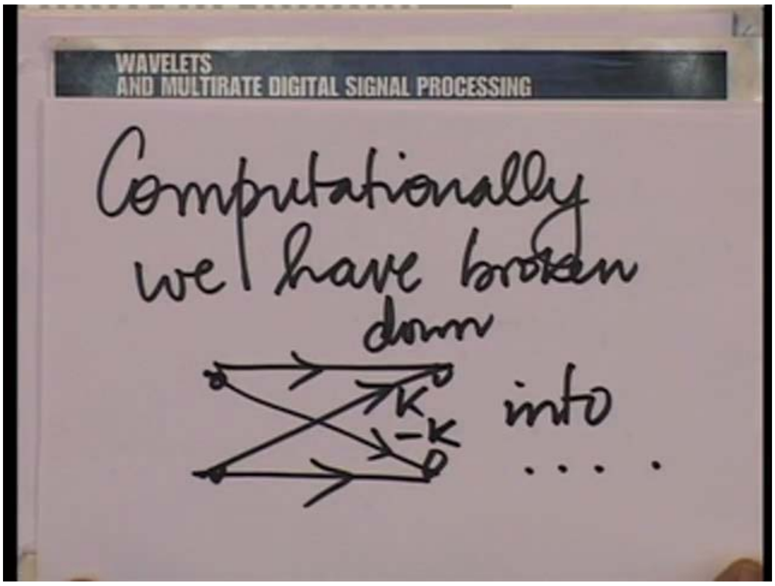



(Refer Slide Time: 39:42)

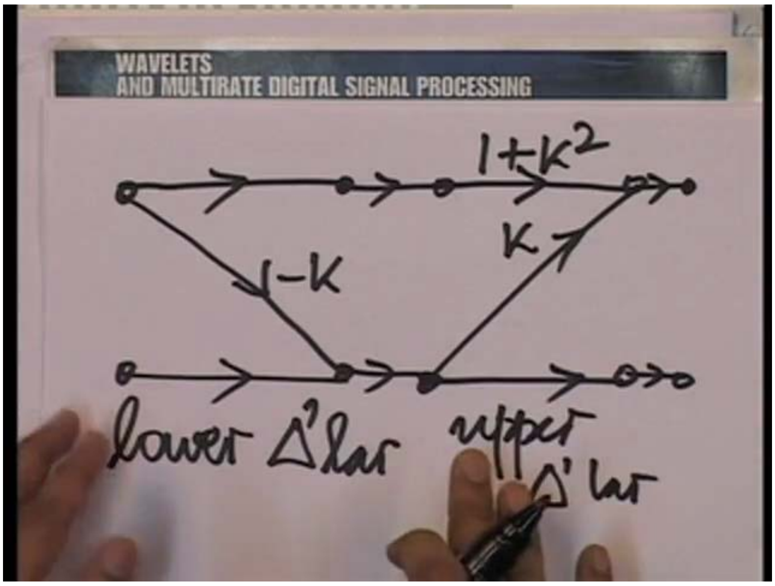



So, we have a plus k times minus k is one which means a is one plus k squared and of course, the others are known b is k c is one and q is minus k. So, overall we have a decomposition of this matrix one k minus k and one into one plus k squared their k here zero and one followed by one on the diagonals here and minus k of diagonal there and. In fact, it is very easy to verify this by multiplication one plus k squared is equal into one minus k squared is indeed one one plus k squared into zero plus k into one is indeed k zero times one plus one times minus k is indeed minus k and zero times zero plus one times one is indeed one this is verified now what we have done here in this decomposition is to break up one lattice stage into even simpler stage of an upper diagonal or upper triangular and lower triangular operation.

(Refer Slide Time: 41:10)

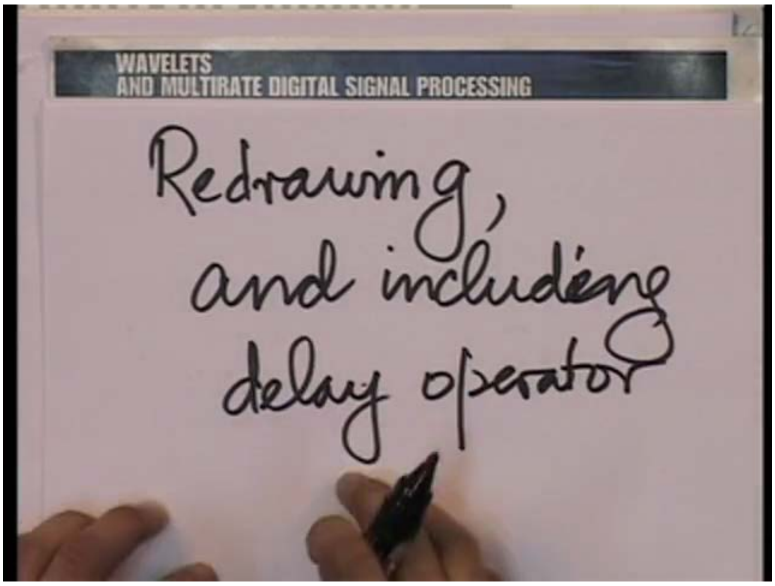



(Refer Slide Time: 41:55)

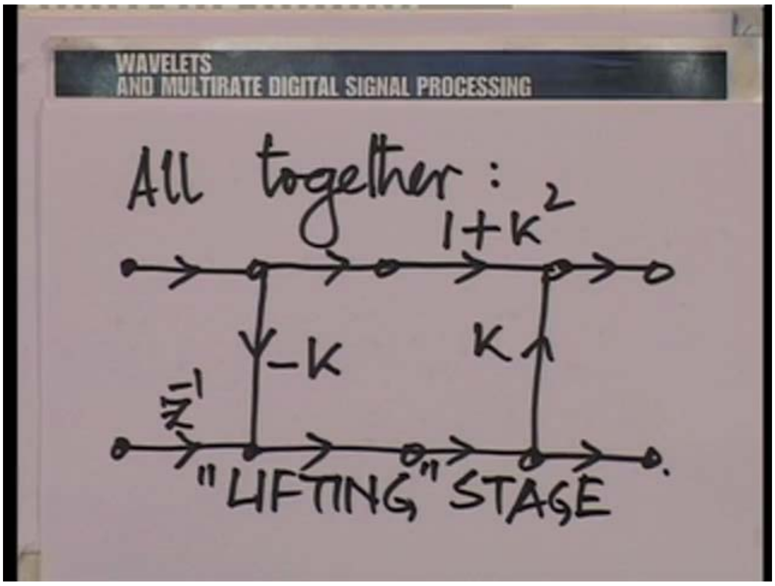



So, what we are saying in a fact is that one lattice stage computationally we have broken down this criss cross operation into a cascade of two operations. So, in the first operation we have one zero; obviously, this matrix operates first. So, one zero minus k one essentially meaning one zero is for the upper one this course as it is minus k one means minus k there and one here. So, the lower triangular part followed by the upper triangular part. So, if you wish I will just put a mark of separation here and the upper triangular part I have one plus k squared k zero and one zero and one essentially mean just this as it is to get this the upper row means one plus k squared mean and k there this is the upper triangular part now you know just for neatness I mean this is just for neatness we could make these lines vertical.

(Refer Slide Time: 43:25)

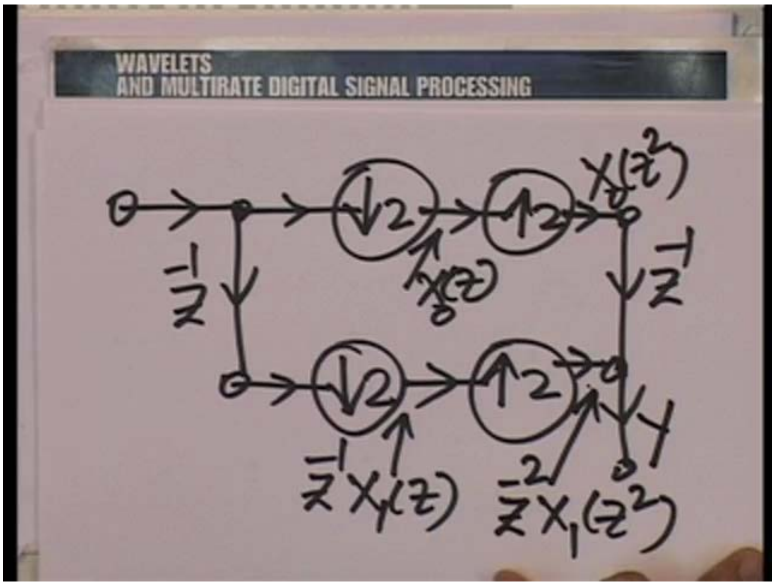



(Refer Slide Time: 43:57)

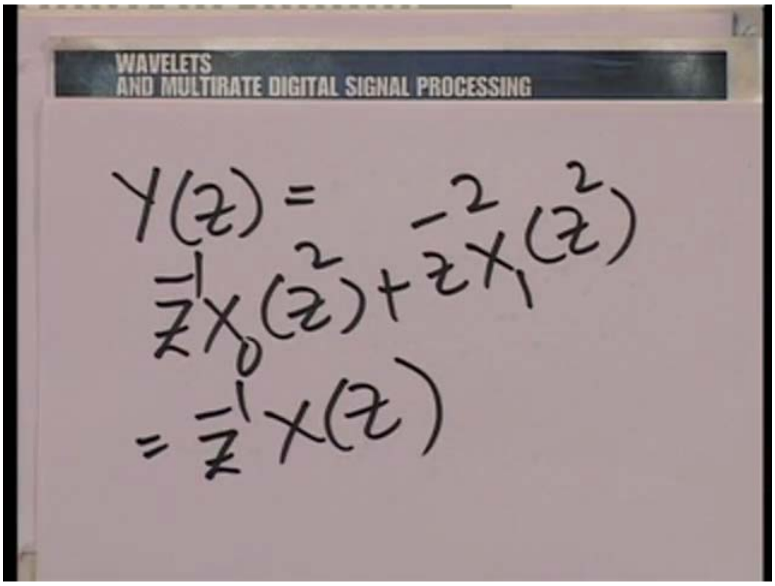



So, we will just redraw it that and also include that delay. So, if you remember we had a delayed before this let us keep the delay as it is. In fact, you know after all we are making a delay and then we are bringing it as it is here remain might is bring that delay here what I mean is all together I have the following a delay this going as it is, but going with a minus k here to produce this followed by this going with a one plus k squared there this going with a factor k to add to this and this going just as it is to produce the lower branch what we have done here is to redraw the structure in a way that makes it very clear that at one time we are essentially just storing one combination operation and this is the central idea behind what is called the lifting stage. In fact, this is called a lifting stage you know the idea of lifting essentially means to lift form no transform to a meaningful transform.

(Refer Slide Time: 45:50)

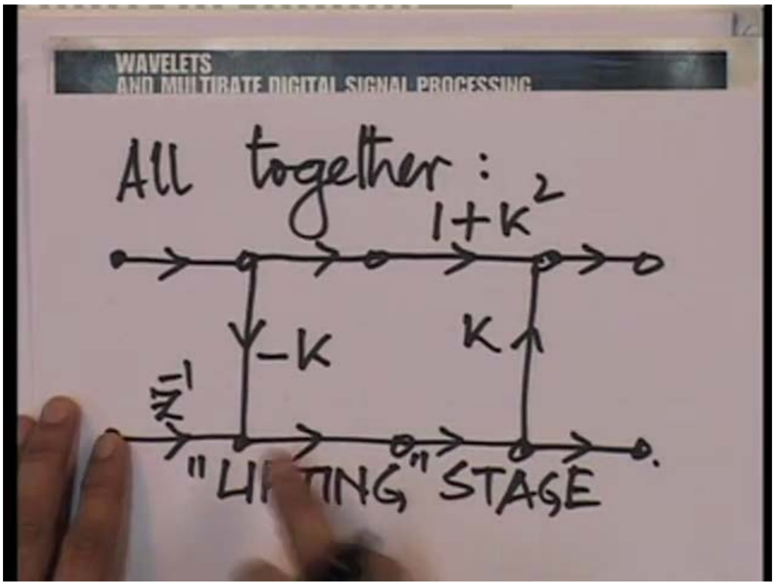


(Refer Slide Time: 47:20)




So. In fact, suppose you where to take just the outputs of the down samplers you have this here and the down samplers here if you where just two up sample this and then recombine. So, I could just put a z inverse there what would I get it is very easy to see that what we get here at this point is x zero z where x zero z has been explained earlier what we get here is z inverse x one z what we get here is this with z replace by z squared what we get here is this with z replaced by z squared. So, all in all what I get here is essentially x zero of z squared what I am get here is z race the minus two times x one z squared and if add this two what I get here let me call this point y is y z equal to z inverse x zeros z squared plus z race the minus two x one z squared which is easily seen to be z inverse times x z the input reconstruct it with a delay of one sample. So, in some sense this is what we have here is what is called a lazy wavelet transform it does nothing at all if I had no lattice stages at all I would have a structure like this. So, from a structure which does nothing at all we are building up stage by stage to a structure which has a meaningful frequency response each of these lower and upper triangular forms essentially built to a better and better frequency response. So, it lifts that wavelets transform which does a great deal both in time and frequency that is why we use that term lifting that is one interpretation of the word lifting.

(Refer Slide Time: 48:55)




So, in the little people of talk to about lifting implementation lift one step at a time and what we had in this structure here is essentially two lifting stages the first time you lift by combining the upper with the lower to modify the lower in the second step we combine the lower with the upper to modify the upper and by alternate improvements are lifting of the lower and then the upper we improve the quality of the wavelets transform step by step what we have built here has been called a lifting structure in the context of discrete variable transforms now what I shown you here is a mechanism to obtained a lifting structure for an orthogonal wavelet transform one can also build corresponding lifting structures for by orthogonal transforms. So, for example, it is the lifting structure which is recommended for implementation in the j five three filter bank which we had described in elaborate detail of few lectures before are this. In fact, that lifting implementation is also possible for many other such by orthogonal filter banks and one of the reason why those particular filter banks five three and nine seven have been chosen in j beg two thousand contexts is because they are communible to lifting a lifting implementation implementation with very elementary operation. In fact, a part of the recommendation in j peg two thousand is to implement the filter bank with lifting structure because of it is computational efficiency as the part of the process of building this lifting implementation we also seen the idea of polyphase matrices essentially matrix operators on the polyphase components and we can see that essentially an analysis filter bank is compose it or is a total matrix operator two cross two matrix operator in the z domain on the polyphase components of the input similarly the synthesis part can also be thought of two cross two operator on the synthesing part to get back the polyphase components of the input or other to construct the polyphase components of output starting from the outputs of the analysis polyphase operator we shall build further on the ideas of polyphase decomposition and polyphase matrices in some subsequent lectures

Thank you.