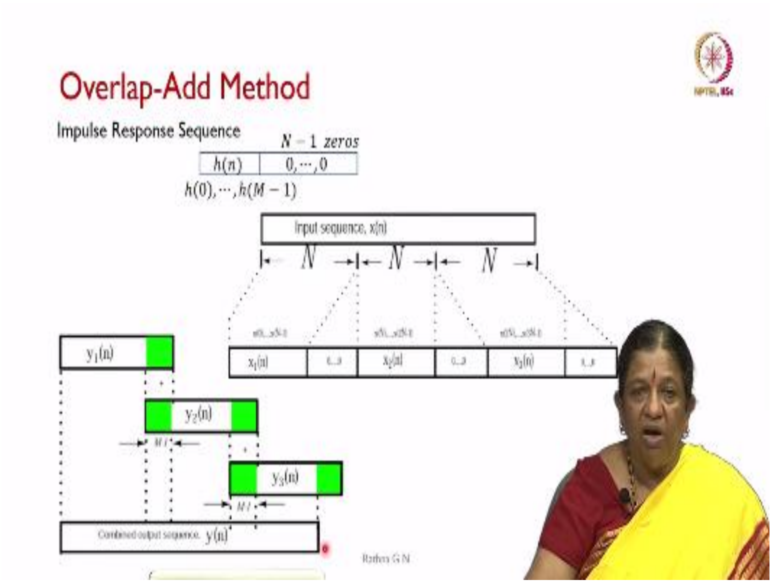
Real – Time Digital Signal Processing
Prof. Rathna G N
Department of Electrical Engineering
Indian Institute of Science - Bengaluru

Lecture – 29
Overlap Save Method

Welcome Back to the real time digital signal processing course so last class we discussed about overlap add method. So, we will take an example in this class and then continue with overlap save method. This is what, what we have discussed overlap add method for continuous signal so, how we can take care of doing FFT for it. So, today, we will take up an example and then see how it is going to work if you have still doubt.

(Refer Slide Time: 00:52)



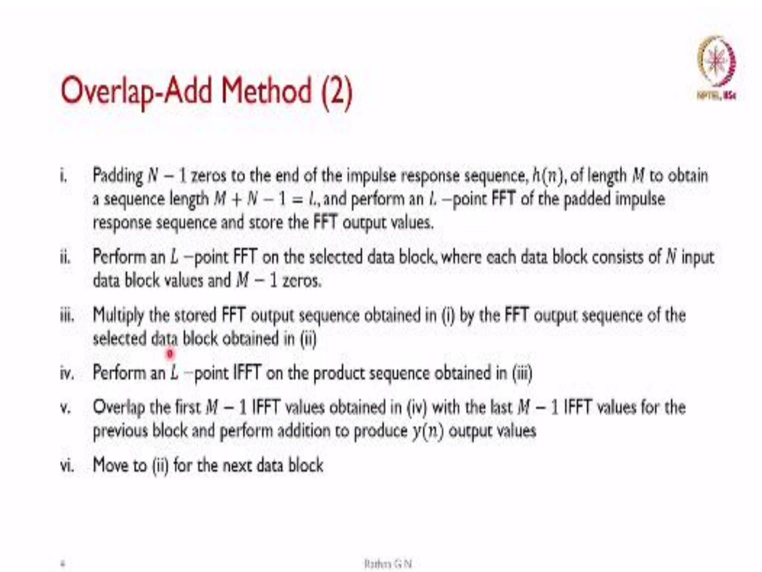
So, this is the procedure what we are going to follow for overlap say add method, this is the impulse response of the sequence that is we call it as $h(n)$. So, $h(n)$ has 0 to M length, and then n is the length of this signal if you assume so we will be adding N - 1 zeros to this. So, this is the what we call it as to make it power of 2 that is N length for our FFT. So, we are going to pad zeros here. So, then we will be considering all N length what we call it as one is the impulse response as well as the input signal as you can see it is more than N length.

So, this can be continuous. So, here for example, we have taken 3 N length in this case. So, we will be bifurcating into N length of sequences of input sequence also, then what we do this is $x(0)$

to $x(N-1)$, we call that as $x_1(n)$ and a pad, this one with zeros. And then the other N samples will be taking types $x(n)$ to $x_2(N-1)$, and then pad with 0s, and then the third and then so on what we will be doing it so in the end, we will be adding 0s as you can see what I will be getting it when I convolve $x_1(n)$ with $h(n)$ basically, we will be getting $y_1(n)$.

So, then what we have is zeros here, the complete this is going to be added to the M - 1 sequence, output of $y_1(n)$ here. And this is $y_2(n)$ next, our M – 1 will be getting added with the $y_3(n)$ in this case, because only we are going to do 3 of it to make it clear for you, but it can happen continuously. So, we add this and then this is our y of n. As you can see here there is a addition sign here, this is getting added with the previous one, and this is the last one which will be discarding it. So, our $y(n)$ will be of this length.

(Refer Slide Time: 03:24)



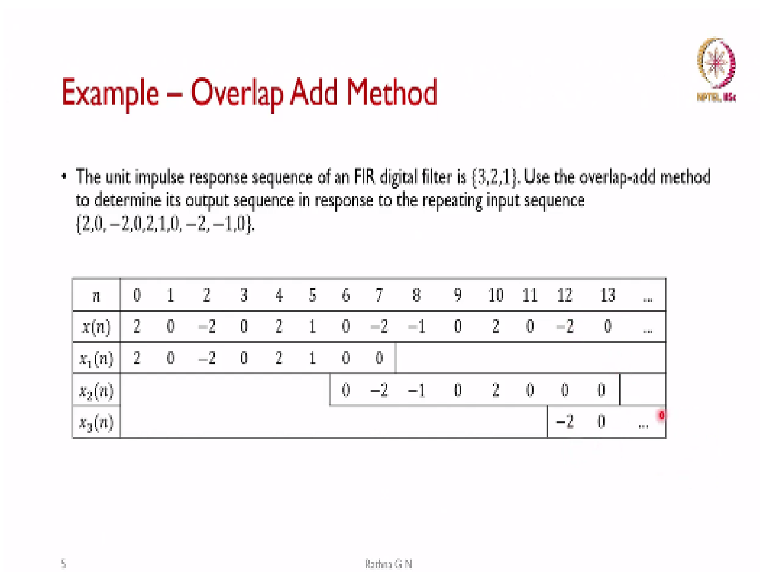
So, to make it clear, so we will take an example in a while. So, how this overlap add method, although we have discussed in the last class, so we will just see that whatever the previous thing I have explained how it is going to work. This is padding N – 1 zeros to the end of the impulse response sequence $h(n)$ of length M to obtain a sequence $M+N-1=L$, and perform L point FFT of the padded impulse response sequences and store the FFT output values.

Then we will be performing L point FFT on the selected data block where each data block consists of N input data block values and M - 1 zeros. So, I think we will have N input data and M - 1 zeros to make it L point sequence which will be taking FFT of it, then multiply the stored FFT output sequences that is because we are doing the filtering basically. So, take the FFT of the impulse response take the FFT of the input sequence then do the multiplication as we know in the frequency domain 2 FFT are going to be multiplied, which is convolution in the time domain.

So, we will be obtaining in one by the FFT output sequence and selected data block obtained from 2. So, performing L point IFFT, so we have got the result so we will be taking inverse fourier transform on the output, and the product sequence obtained in 3 here what we will be doing the IFFT, then what we will do is overlap the first M – 1 FFT values, obtained in those 4 with the last M – 1 IFFT values for the previous block, and then perform addition to produce y of n output values.

And then we will be moving back to this stage, because we need not have to as we discussed in the computation complexity FFT, because once we have done the FFT of our impulse response, we need not have to redo the thing only for the next set of input blocks, we will be taking FFT, so we will be moving to the next data block, that is we will be performing it in the loop.

(Refer Slide Time: 06:02)



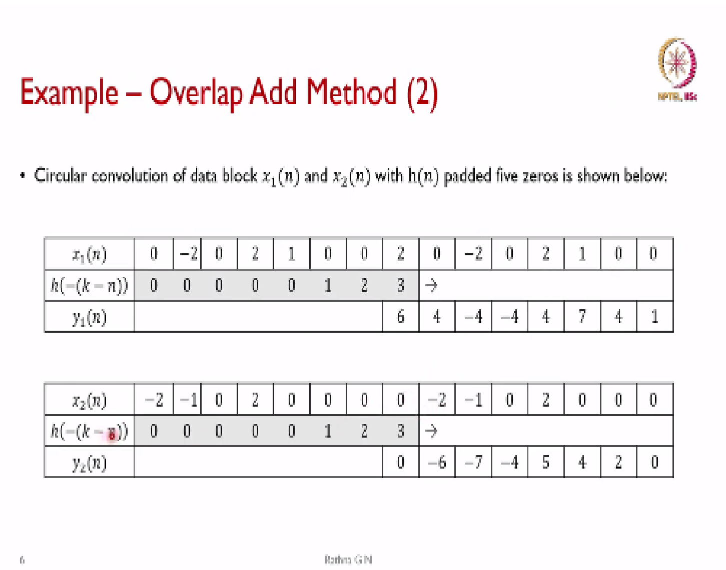
n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	...
$x(n)$	2	0	−2	0	2	1	0	−2	−1	0	2	0	−2	0	...
$x_1(n)$	2	0	−2	0	2	1	0	0							
$x_2(n)$							0	−2	−1	0	2	0	0	0	
$x_3(n)$													−2	0	...

So, we will take up an example. So, which will make it clear to you, so, in this case, M point filter it is 3 bits, what we have take a sorry, 3 point length, what we have taken the thing, so it is 3 2 1 is the impulse response. And we will be using overlap add method to determine the output sequence in response to the repeating input sequence the size what we have chosen is 2, 0, - 2, 0, 2, 1 etcetera as it is shown. So, now, our M is the 3 and then our n what we are going to select as you can see the thing here 2 4 6 and 8.

So, x 1 of n, so we will be first considering it with 0s at the what is it we call it as in the end of our sequence, so, we are considering our N point as 6 in this case, M is 3. So, 6 + 3 is 9 - 1 will be L so, which is equivalent to 8 in this case. So, this is how we compute our L, M and then N basically. So, from here, what we are going to do is our now $x_2(n)$ is as you can see in the figure here. So, we will be starting from here to here after padding with 0s. So, in this case 0, - 2 and then we are taking the rest of the signals here up to 0 here.

Then we pad again with 2 0s M - 1 0s, which is equal into 2 0s what we are adding it here and then next sequence what will have it is - 2 and 0 in this case and so on. So, this is how we have calculated our L, M and then N values.

(Refer Slide Time: 08:20)



$x_1(n)$	0	−2	0	2	1	0	0	2	0	−2	0	2	1	0	0
$h(-(k-n))$	0	0	0	0	0	1	2	3	→						
$y_1(n)$								6	4	−4	−4	4	7	4	1

$x_2(n)$	−2	−1	0	2	0	0	0	0	−2	−1	0	2	0	0	0
$h(-(k-n))$	0	0	0	0	0	1	2	3	→						
$y_2(n)$								0	−6	−7	−4	5	4	2	0

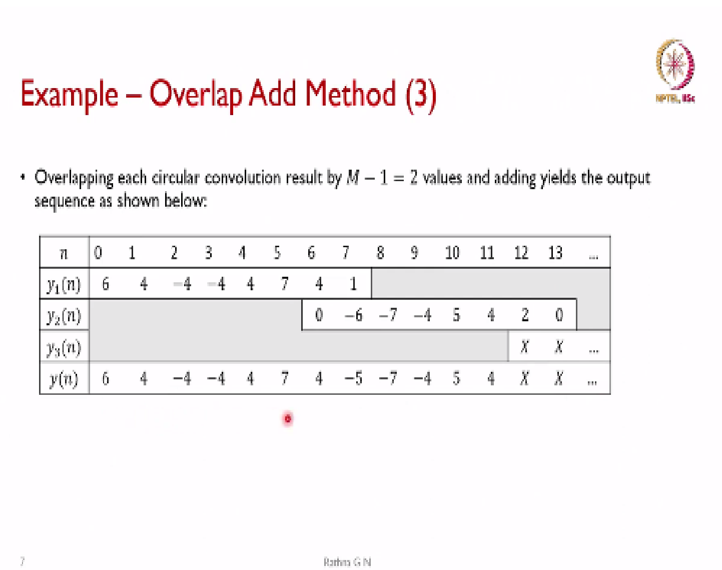
So, coming with the thing, how we are going to continue? Now we have $x_1(n)$, $x_2(n)$. So, we will be convolving with our h of n which is padded with 5 0s. So, as it is shown here. So, and then we have to take the what will be calling it as $h(-(k-n))$ values what will be taking it so, what we have been given is 3 2 1 and then you will be doing the reversal of the sequence. So, first these are the 5 0s then 1 2 3 what will be taking it then this is going to be convolve with $x_1(n)$ what we have chosen.

So, now our $y_1(n)$ is going to be as it is seen. So, we have it 3 into 2 which is going to be 6. So, some previous values what you can have it because you will be moving the sequence as it is shown it is going to be move to the right. In the next step what you will be getting 2 into 2 is 4. So, then you will be getting 3 into -2. So, which is going to be -6 and then here you are going to have 2 into 1. So, this is what the value which is going to come here in the second clock cycle.

So, -6 + 2 is going to be -4 so on you will be till the all the values are have been computed. So, this is our $y_1(n)$. So, this is how we will be doing your circular convolution, then now, $x_2(n)$ is the sequence what we have it from the previous what we have taken the thing. So, this is -2 – 1 0 2 and then 0, 0, then 2 padded with 2 0s, and then this will be going in the forward direction as you can see it here. So, the first one will be 0 output, so when you move towards your right because, as you know the circular convolution, so we will be repeating those values.

So, this is -2 into our 3, which is -6 and so on you can compute it so, the output of $y_2(n)$ is given here.

(Refer Slide Time: 10:51)



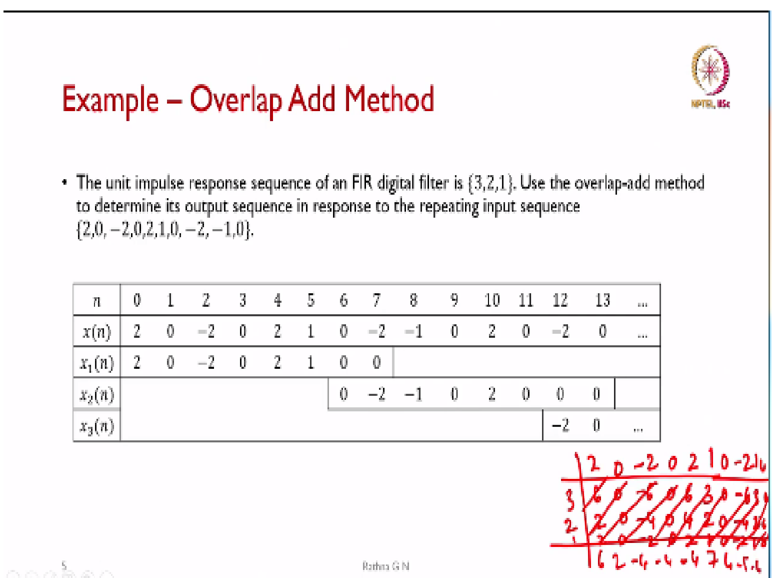
n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	...
$y_1(n)$	6	4	−4	−4	4	7	4	1							
$y_2(n)$							0	−6	−7	−4	5	4	2	0	
$y_3(n)$													X	X	...
$y(n)$	6	4	−4	−4	4	7	4	−5	−7	−4	5	4	X	X	...

Then next is how we are going to use this overlap add method. So, we know that convolution result by M – 1 = 2 values and adding yields the output sequence as shown below. So, that is this is my $y_1(n)$ and next $y_2(n)$ is going to be aligned with these 2 that is what it says 2 values have to be over lapped from $y_2(n)$. And then we have to add these 2 and then our $y_3(n)$ because only we have 2 and then 0 and then later on it is not defined, so we will be calling it as X, X.

So, if we have some more values, then I had to compute my $y_3(n)$ in the same way as y_1 and then y_2 take those values and then put it here. Now, what will be the final sequence so it is 6, 4, -4, -4, 4, 7. And then 4 + 0 is 4, -6 + 1 is -5, and then these sequences will be repeating it and then after that I am not bothered I can put it as X X X. So, if you see your convolution, what output you are going to get it whether it is equal into this or not one can look at it using the overlap add method.

To show that it is correct, what we have to copy is copy the sequence and then copy your 3 2 1 and then do the normal convolution. If you are interested it, that is what we will put it as show with the thing.

(Refer Slide Time: 12:35)



So, normal all of you know this convolution pattern so, this is what you have it is 2, 0, -2, 0, 2, 1, 0, -2, -1 and then last 0. So, this is what so, what we have is 3 2 1 are our sequence, I can put a line here 6 0 -6 0 6 3 0 -6 and then one what I have it 3 and then 0, last one. So, here it is going to be

2, 0, -4, 0, 4, 2, 0, -4, 2 and then 0. The last one will be 2 0 -2 0 2 1 0 -2 1 and then 0. So, you know that this is the way what you will be adding up in normal linear convolution.

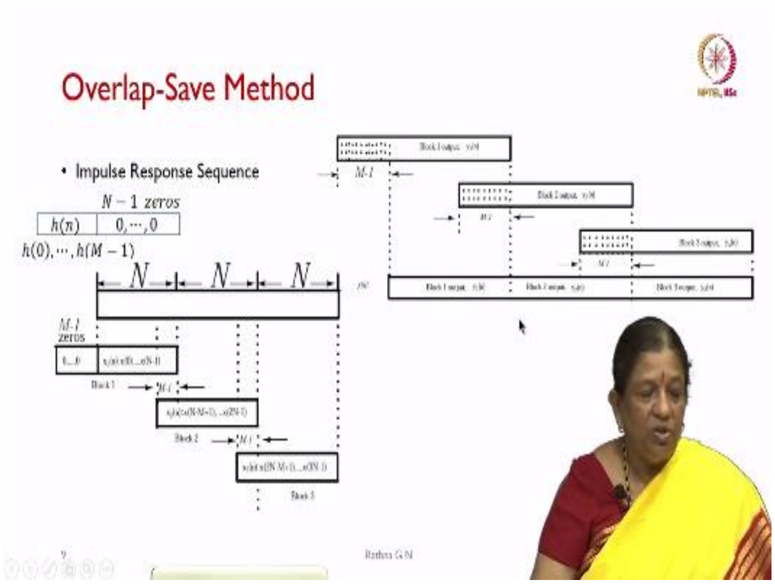
So, you can do that and then see whether you will be getting the whatever you have got the output correctly or not. So, these 3 what I have to do the thing, so, first one is 6, second one is 2, -6 + 2 is -4, and then -4 and then 6 - 2 is again -4. So, then what we have it is 4 + 3 is 7, 2 + 2 is going to be 4. So, I have -1 -5 and then -4 and you can compute the thing. So, go back and then check whether our output what we have got it is correct or not here.

So, this is how will you can cross verify and then see whether your convolution output is using this overlap add method because this is a simple length of it, what it has taken to work it out by hand, so, that when you write your code in MATLAB or in C, you can verify it and then for larger $x(n)$ sequence and then whatever filtering what you have to do it, you can use that and then run it. So, in the lab will demonstrate that whether it is going to work or not, this is one of the way of computing the long sequence using overlap add method.

(Refer Slide Time: 15:26)

So, the next one, what we will see is overlap save method. So, here it is, this one is called also overlap discard method will, please hold on a while, while we are there, we have added the whatever M - 1 0s we have added here, M - 1 data, we are going to discard it, that is why either we can call it as overlap save or overlap discard method.

(Refer Slide Time: 15:53)

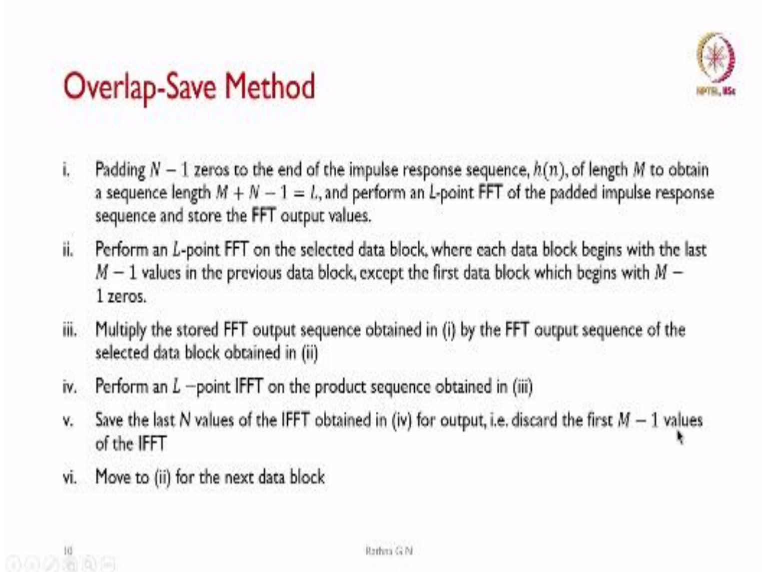


So, how this method is going to work same as this we have impulse response. So, we have $h(0)$ to $h(M-1)$ is the M length the impulse this thing sequences and then next will be padding with N – 1 0s to make it an L length sequence now we have N length input $x(n)$ what we are going to consider and the next all these are again just like previous overlap add method. So, we have N length sequences what we have considered then what we are going to do, so, for the first one we are going to add M - 1 zeros to make it N length sequence.

And then for the next one, we are going to have M - 1 samples from the previous one what will be adding for the current length of the sequence. So, that is we call it as block 2 this is the first block, block 2 and then so on block 3 and then if we have other this thing inputs, it will be going on that way then how it is going to work. So, you can see what will be our output this is M - 1 will be discarding it and then block 1 output after can this thing the multiplication will be working on $y_1(n)$.

That is $x(k) \cdot h(k) = y_1(n)$ then taking IFFT $y_1(n)$ is going to come and here it is again, we are going to discard M - 1 in this sequence, and then take the rest of the block 2 as $y_2(n)$. And the other M - 1, we are going to discard this output and then take the rest of the $y_3(n)$ and we will be concatenating these 3 blocks y_1, y_2 and y_3 this will be our output. So, you would be wondering how this is going to work we will take the same example and run the case in a while.

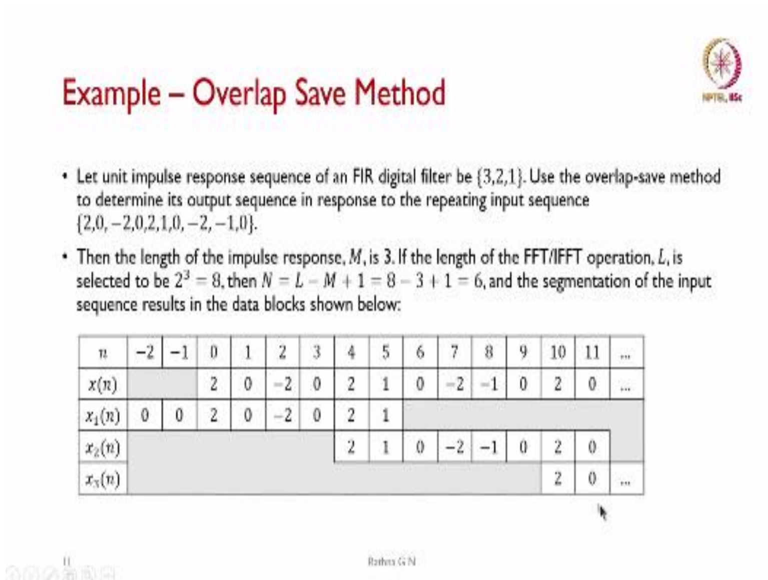
(Refer Slide Time: 18:02)



So, what is the procedure for it that is for the overlap save method, I had to pad same as overlap pad, pad N – 1 zeros to our filter length to make it length L by making N – 1 zeros to M length impulse sequences to make it as L length then to the L point FFT. And then same thing with our input how we are going to select here so selected data block where each data block begins with the last M - 1 values in the previous data block except the first data block which begins with M - 1 zeros. That is what, what I showed you in the previous slide.

Now do the multiplication of these 2 FFT and then take by the FFT of this thing block with respect to this, then perform L point IFFT to the product sequence which is obtained in 3 and save the last N values of IFFT obtain from this place, and then discard the first M – 1 values of the IFFT. So then, we will be moving back to calculate the next sequences.

(Refer Slide Time: 19:27)

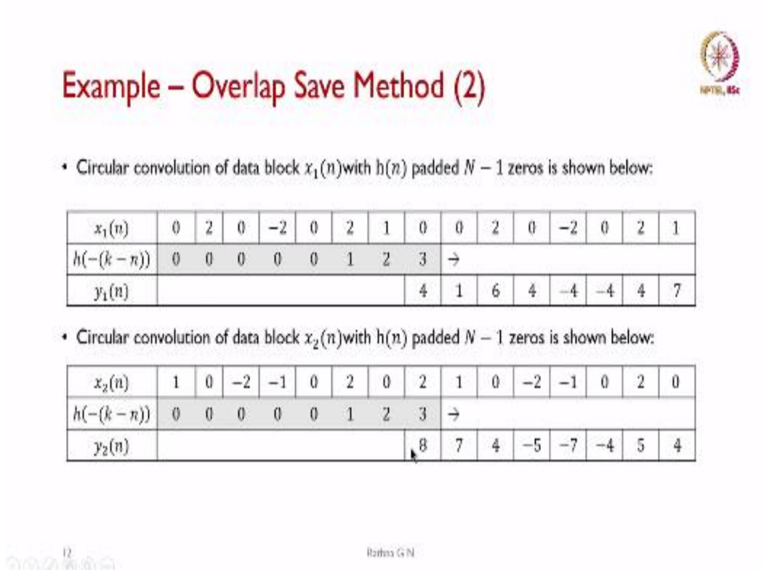


So, we will see how it is going to work with the same example what we have taken. So, the impulse response for the FIR filter order is M which is equal to 3 the values are same thing 3 2 1 and then input sequence is same what we have assumed in the previous case, now, you will be seeing that your M is 3. If the length of the FFT or IFFT operation L is selected based on this that is 2 power 3 which is equal to 8 then N becomes l - M + 1, which is nothing but 6. This is how we arrived at 6 in the previous case also.

And the segmentation of the input sequence results in the data blocks shown in this case, that is n what we have named it -2 -1 because we need M – 1 zeros now, which is going to be padded before the input sequence, so, we will be naming it as -2 -1and then the input starts from zero onwards. So, our input is 2 0 -2. And then up to here, what we have it as x of n, this is a complete x of n and $x_1(n)$ is going to be up to here that is padded with 2 zeros. And then we will be taking 6 sequences as you can see here 2 0 -2 0 2 and 1 are the sequences what it has been assumed.

Now, what will be our $x_2(n)$ we set M - 1 previous samples from the x(n) what we have to take it so, in this case, we have ended $x_1(n)$ here. So, previous to that 2 samples means this 2 and 1 are going to be repeated in our $x_2(n)$ and rest of the 6 samples are going to be from our x(n) here, so, you will be pushing it down. Same thing with the $x_3(n)$ our last 2 and then we will be repeating it and then goes for further.

(Refer Slide Time: 21:36)



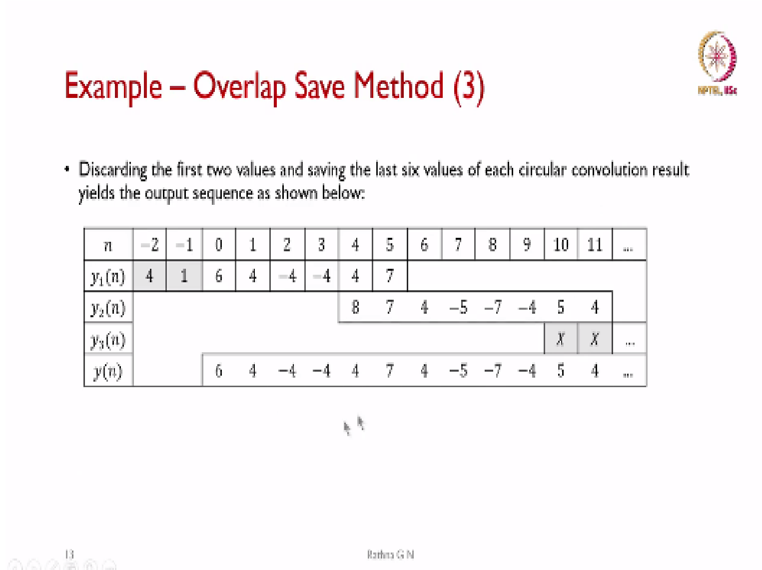
$x_1(n)$	0	2	0	−2	0	2	1	0	0	2	0	−2	0	2	1
$h(-(k-n))$	0	0	0	0	0	1	2	3	→						
$y_1(n)$								4	1	6	4	−4	−4	4	7

$x_2(n)$	1	0	−2	−1	0	2	0	2	1	0	−2	−1	0	2	0
$h(-(k-n))$	0	0	0	0	0	1	2	3	→						
$y_2(n)$								8	7	4	−5	−7	−4	5	4

Now, we will see its operation first is we have taken $x_1(n)$. So, we are doing the circular convolution here. This is our $x_1(n)$ and after that as you can see it is repeated for the next length also and our impulse response $h(-(k-n))$ is given by this we have padded here also with 5 0s and then you have taken the reverse the sequence which is going to be 1 2 3 start the computation here. So, what is it initially you will be seeing that this is 2 into 1 + 2 into 1 which is going to give us 4 and then shift by 1 bit and then start computing it.

So, the next will be 1 and so, on compute till here, then, next is $x_2(n)$. So, we said that the sample what we had was from the previous one what it was repeated, you will be seeing that 2 and then 1 here and then you will be having the input what it has been taken from the input sequence. So, now, same thing with $h(-(k-n))$ so, you will be reversing it and then taking it and then you will be doing the circular convolution of the 2 sequences. So, you will be seeing that the resultant $y_2(n)$ is 8 7 4 -3 -7 -4 and 5 and then 4.

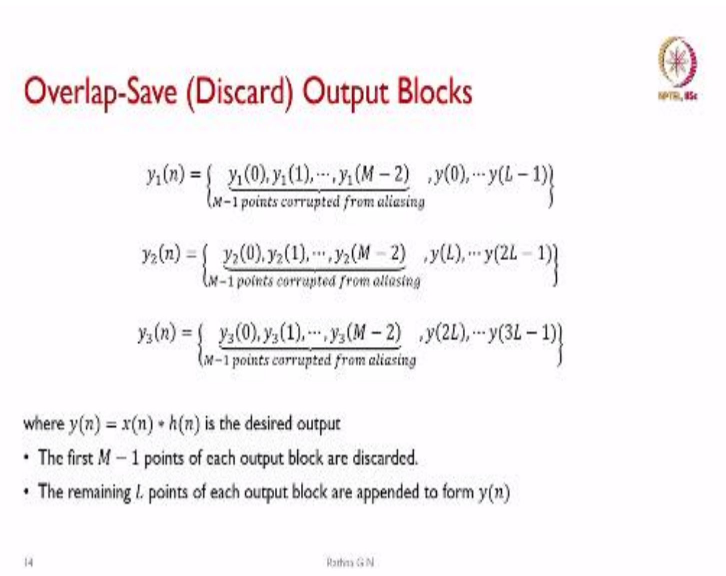
(Refer Slide Time: 23:21)



Now, the next step is we have to calculate $y_1(n)$ is given $y_2(n)$ and $y_3(n)$ last 2, we can now ignore them, that is what, what it will be. So, you are putting under y_1, y_2, so, we are going to discard these 2 values after the thing doing inverse FFT. And then this is what it will be resulting is 6 4 -4 -4. And then you are going to discard these 2 values from $y_2(n)$ and then put 4 7 from here, and then continue with whatever data you are going to get from $y_2(n)$ after discarding these 2 samples.

So, you were seeing that both overlap save and then overlap add method. So, works same as with regular linear convolution. So, we are getting the same results. So, this is the way how overlap save method works.

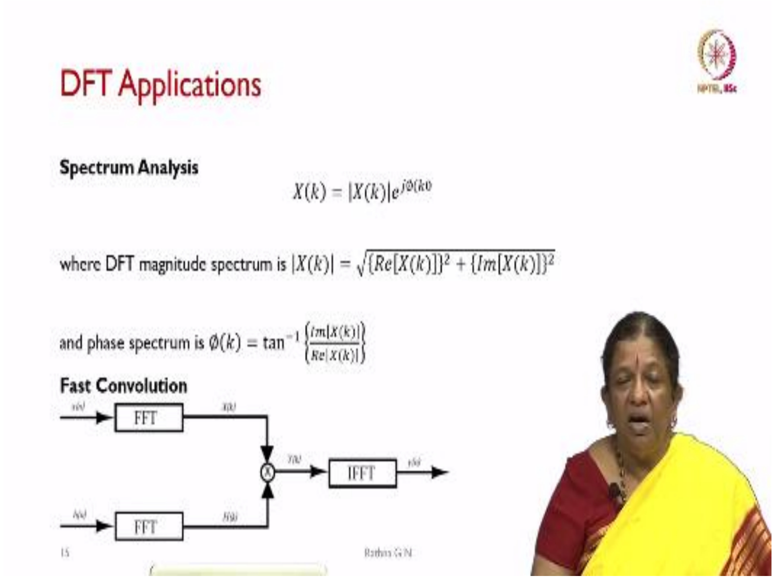
(Refer Slide Time: 24:28)



So, you will be seeing that how the discard output blocks is going to happen with respect to y_1, y_2, y_3, is given in this case. So, $y_1(0), y_1(1), \cdots, y_1(M-2)$ what you will be discarding them M - 1 points and then you will be considering only this points and then these things, you will be discarding. So, this is a equal into our regular that is linear convolution with $x(n)$ with $h(n)$. So, this is what the desired output and first M - 1 points of each output block are discarded and the remaining L points of each block are appended to form the $y(n)$.

So, this covers our both overlap add and then save method to compute FFT for a long length sequence. So, one can ask why we have to have a overlap. So, without overlap I have not taken the example in this case, one can work it out using MATLAB so, what is the thing is going to happen? So, you will have the discontinuity, as you know that there will be if it is speech signal if time permits, we will show you in the next class a demo of it without overlapping how the signal looks like.

(Refer Slide Time: 26:02)

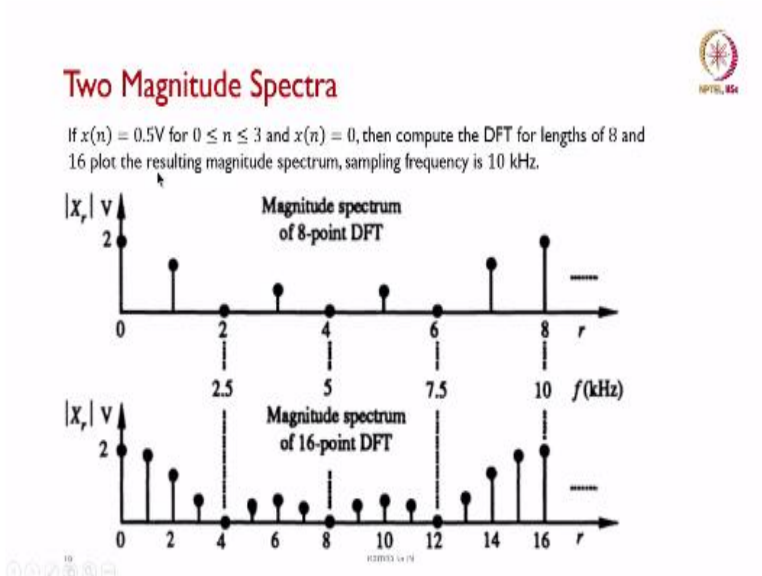


So, now what are the applications of DFT so, the first one as I told in the previous class that it is spectrum analysis. So, what do we mean by that, that is $X(k)$ is nothing but our $|X(k)|$ into this is the phase part of it. So, we will be taking the magnitude spectrum is given by that is $|X(k)|$ is nothing but a $\sqrt{\{Re[X(k)]\}^2 + \{Im[X(k)]\}^2}$ what will be computing it. So, this we have already seen in MATLAB as well as in CCS, what will be our magnitude spectrum is going to be.

And next is the phase spectrum one wants to have it so, which is going to be $\tan^{-1}\left\{\frac{Im[X(k)]}{Re[X(k)]}\right\}$. So, in the example in the last class, we computed for DFT what will be the angle and we plotted manually both the phase and then magnitude spectrum. So, now, one more application as we call it of the DFT is fast convolution. So, as we computed in the last class, what will be the computation time for FFT calculation, and then how this can be implemented to do a fast convolution.

That is $x(n)$ is we take the FFT, which is $X(k)$ and then impulse response are B coefficients in FIR filter, we can take the FFT of it, pre computed, we will do that and then do the multiplication. So, we have considered the complete computation time and we have to do IFFT and then get the $y(n)$. So, compared to the direct DFT or direct convolution, so, how we were able to achieve almost 5 times the computation speed compared to the normal one that was shown in the last class.

(Refer Slide Time: 28:21)

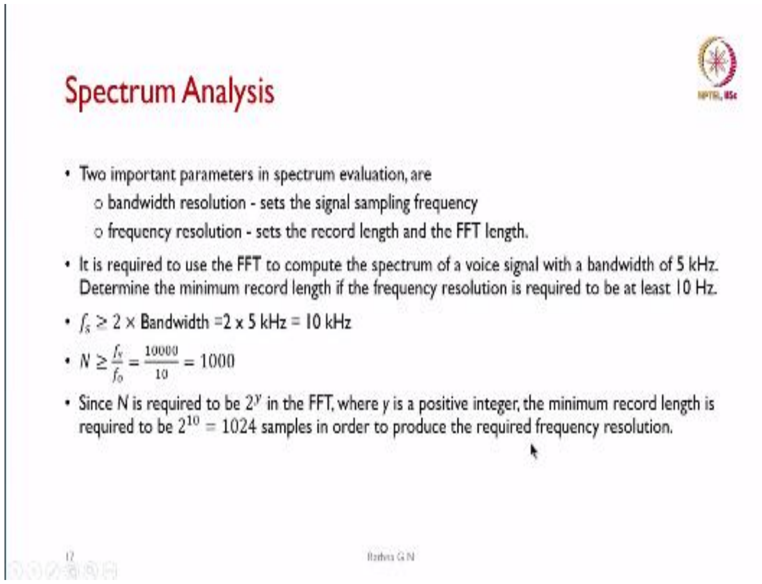


So, then know to check the thing how to calculate all though we have done it in the last class, we will see how to plot our magnitude spectra with an example. So, the example is $x(n) = 0.5\text{V}$ for $0 \le n \le 3$ and $x(n) = 0$ in other places, then compute the DFT for lengths of 8 and then 16 and plot the resulting magnitude spectrum sampling frequency in 10 kilohertz.

So, the thing is, what we have is, if you calculate manually you can do this and then plot it. So, the magnitude spectrum you will be seeing that the n = 0 will be 2 and then which comes to 1 0 and then 0.5 and then you will be seeing 0.5 and then you have 1 and then 2, this is with respect to n = 8. So, the same thing if you do with n = 16 that is what we checked it increasing that is the magnitude spectrum for DFT how it is going to look like so, you will be seeing that few of the samples are in between filled between 0 to 1 in this case because it is twice.

Compared to 8 kilohertz 8 point it is 16 point so, I will be adding one more point in between these 2 signals. So, you will be seeing that between these 2, as you can see that this point 3 has been added. So, at here 1 and this 3 and then you will be seeing all odd values have got added with respect to n = 16. So, you will be seeing when you draw a line in the thing. So, this may be much what is it, we call it as a smoother one to predict your computation using FFT.

(Refer Slide Time: 30:37)

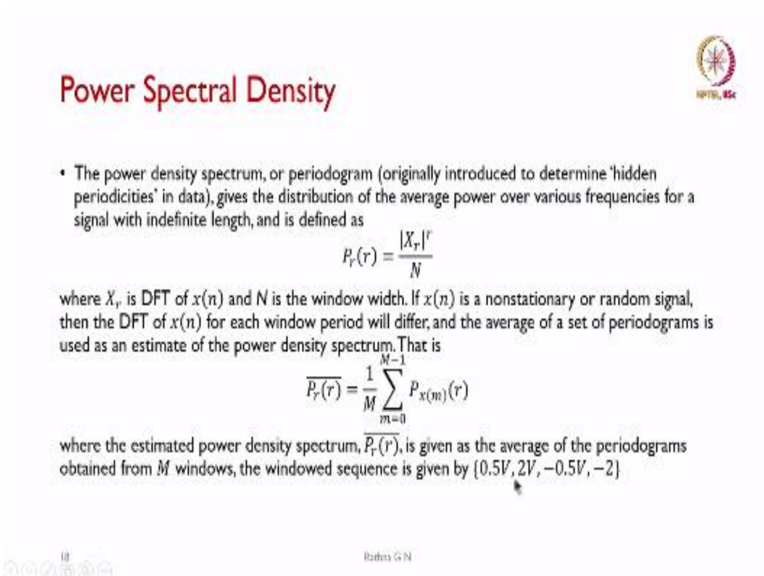


So, now, we will see how we can do the spectrum analysis. So, 2 important parameters in spectrum evaluation are one is the bandwidth resolution, the other one is the frequency resolution. So, the bandwidth resolution sets the signal sampling frequency whereas, our frequency resolution sets the record length and FFT length. So, as an example, it is required to use FFT to compute the spectrum of voice signal with a bandwidth of 5 kilohertz determine the minimum record length if the frequency resolution required to be at least 10 hertz.

So, that means, from sample to sample what we want to have it as 10 hertz, then what is it sampling frequency has to be greater than or equal to twice our bandwidth. So, which restricted sampling frequency is 10 kilohertz, then we will see that what will be our record length N what we have to calculate this should be greater than or equal to or $\frac{f_s}{f_o}$, f_o is a spacing what it has been given is 10 hertz. So, that means to say that 10 kilohertz divided by 10 is going to give us 1000 samples basically.

So, to do our FFT computation, we know that nearest power of 2 what we have to assume it for 1000 it is going to be 1024 that is 2 to the power of 10 is going to be 1024 is the nearest FFT computation, what we have to do it for the signal order to produce the required frequency resolution. So, then we will see that in the previous case. So, now, you will be seeing that instead of 10 hertz, so, you can go back and then check what will be the frequency resolution because we are trying to fix the record length based on it the frequency resolution little bit get modified it may come to 9.9 hertz you can check it up.

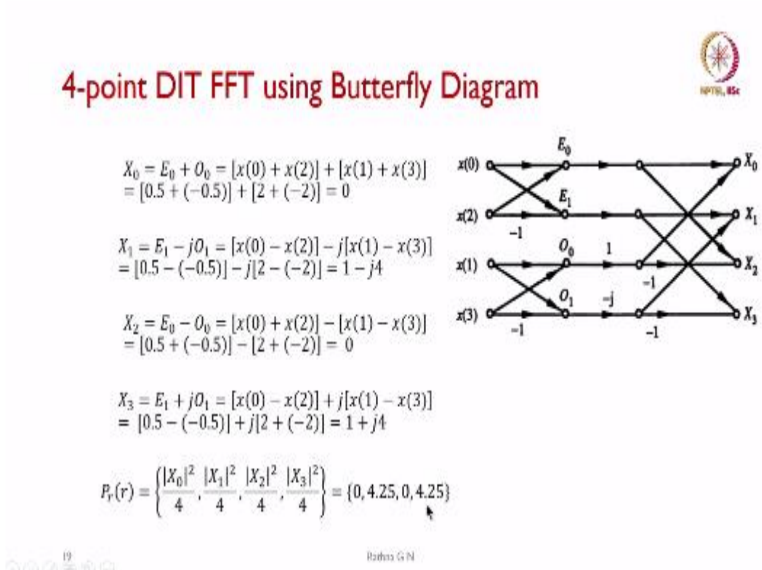
(Refer Slide Time: 32:49)



Next is how to compute our power spectral density. So, we know that power density spectrum or periodogram we call it was originally introduced to determine our hidden periodicities in data. So, gives the distribution of average power or various frequencies for a signal with indefinite length and is defined as p r r is given by the magnitude of X r to the power of r divided by N. So, we know that X_r is our DFT of $x(n)$ and N is the window width, if $x(n)$ is nonstationary random signal then the DFT of $x(n)$ for each window period will differ.

And the average of a set of our periodogram is used as an estimate of the power density spectrum, which is given by $P_r(r)$ of so, we are averaging over M Windows 1 by M times. So, $\sum_{m=0}^{M-1} P_{x(m)}(r)$ where the estimated power density spectrum $P_r(r)$ over the thing what it is represented is given as the average of the periodogram obtained from M windows, the windowed sequences given by example, if you take it 0.5 volts, 2 volts, -0.5 volts and -2, this is our $x(n)$.

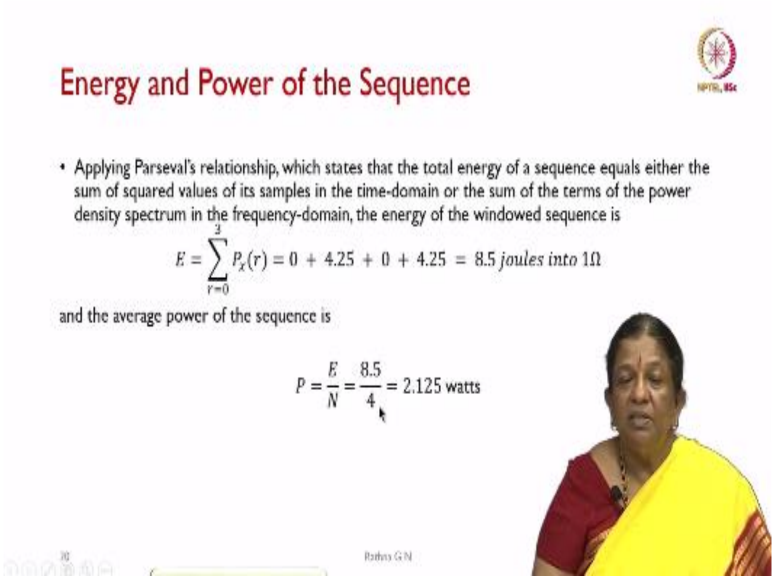
(Refer Slide Time: 34:23)



We will see the example how we will be calculating it. So, you have been given values as – 0.5 2 –0.5 and -2. So, now we will see that if we because 4 point, so we will apply the decimation in time FFT butterfly diagram is shown here. So, if we input the thing, so and then compute our X_0, so, we have worked out this example. So, if you want to see the steps have been given here, so, the first $X_0 = 0$, and then the $X_1 = 1 - j4$. So, third again is 0.

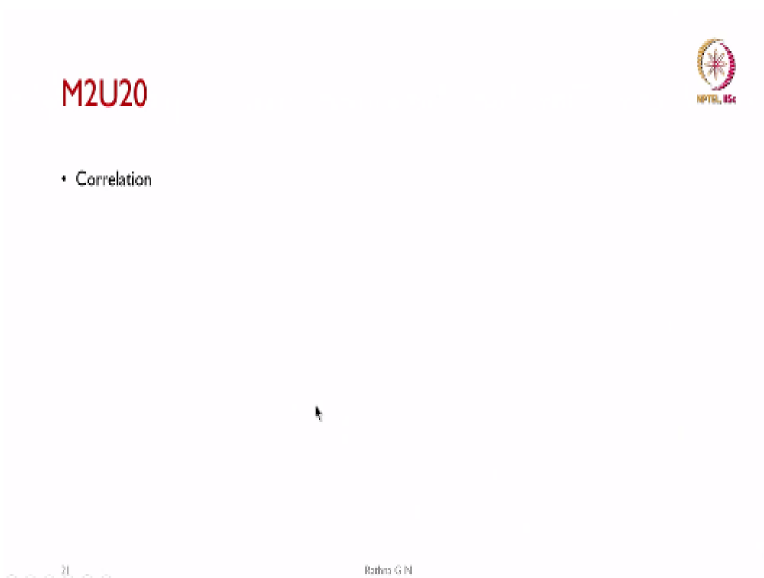
And then we know that X_3 is going to be called conjugate of our X_1 basically, so which is going to be $1 + j4$ or you can compute using our butterfly diagram. So, then how we are going to compute our power spectral density. So, you will be seeing that $\frac{|X_0|^2}{4}$ and then you will be seeing this way, then what happens to the thing, it is 0, 4.25, 0 and then 4.25. So, this is the distribution in the thing.

(Refer Slide Time: 35:44)



To find out the energy and power of the sequence so, we know that possible relationship, what we are going to apply, so for the energy sequence, so which is given by $E = \sum_{r=0}^{3} P_x(r)$, in this case for the example. So, this is $P_x(r)$. So, when you calculate 0 plus add them up, so we will be seeing that it is going to consume 8.5 joules into 1Ω resistor. So that is if you are passing this on a 1Ω resistor, it will consume 8.5 joules and the average power if you want to calculate of the sequence, so you will be calculating energy divided by N basically. So, it is going to be 2.125 watts of power, what this system is going to consume.

(Refer Slide Time: 36:42)



So, this ends our overlap add and then save method and how to compute our power spectral density and then energy of sequence. So, in the next class, we will take a correlation. Thank you.