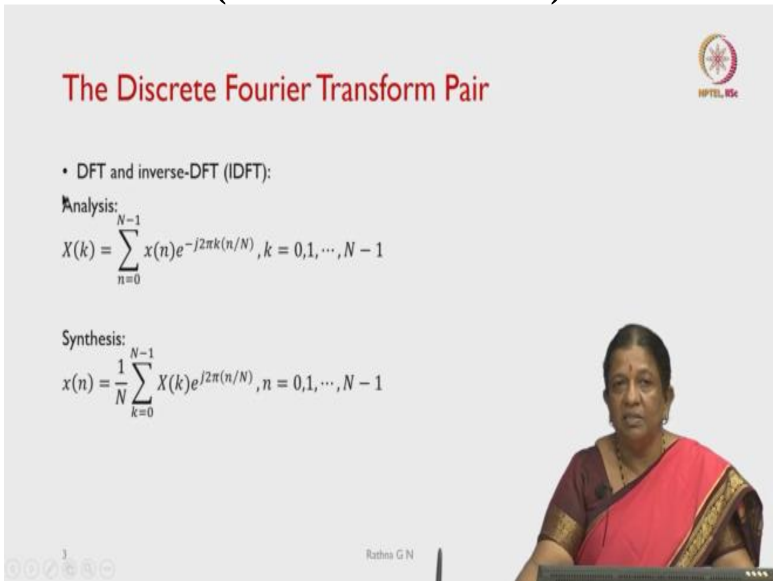
Real-Time Digital Signal Processing

Prof. Rathna G N
Department of Electrical Engineering
Indian Institute of Science - Bengaluru

Lecture - 27
FFT - 3

Welcome back to real time digital signal processing course, today we will discuss about continue with our FFT. So, in the last class, we discussed about FFT for fast computation, how it is going to help us in speeding up our computation from DFT. And then we saw that what are the quantization effects of our FFT in computation and then how we are going to do the scaling.

(Refer Slide Time: 00:57)



The Discrete Fourier Transform Pair

- DFT and inverse-DFT (IDFT):

Analysis:

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi k(n/N)}, k = 0,1,\cdots,N-1$$

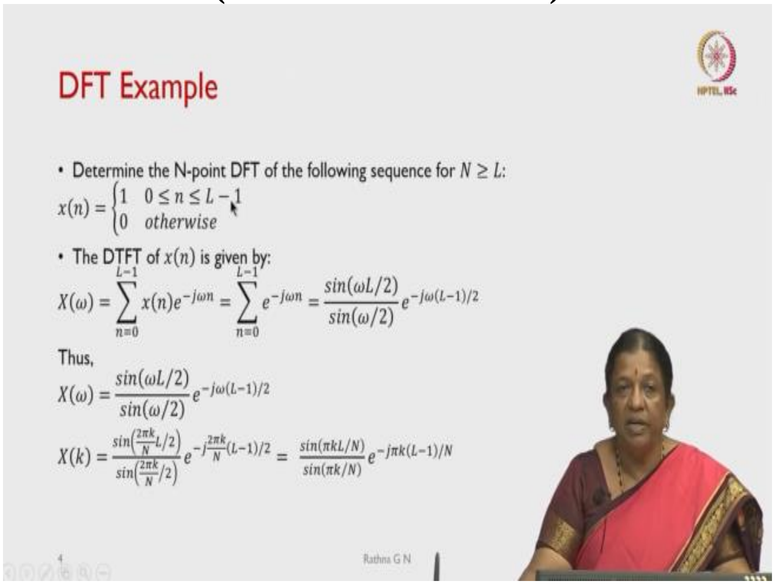
Synthesis:

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k)e^{j2\pi(n/N)}, n = 0,1,\cdots,N-1$$

So, today we will talk about a little bit on now computation aspects of DFT and then FFT with examples. So, we call discrete Fourier transform pair as analysis and synthesis equations. So, analysis equation is given by $X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi k(n/N)}, k = 0,1,\cdots,N-1$. So, we can represent this with that twiddle factors and inverse DFT. As we know it is IDFT, which is given by $x(n)$.

We have the scaling function $\frac{1}{N}\sum_{k=0}^{N-1} X(k)\, W_N^k$, here it is W_N^k, here it is going to W_{N-n}^k what we will be taking it according to the twiddle factors. To derive this $x(n)$, what we can do is we can substitute our $X(k)$ with this equation, and then derive the thing. So, for this derivation that $x(n)$ becomes $x(n)$ and why we have the $\frac{1}{N}$ scaling function also, you can refer to any of the digital signal processing book. So, which will give you how to derive our, that is analysis, substitution in synthesis get our $x(n)$ back.

(Refer Slide Time: 02:25)



DFT Example

- Determine the N-point DFT of the following sequence for $N \geq L$:

$$x(n) = \begin{cases} 1 & 0 \leq n \leq L-1 \\ 0 & otherwise \end{cases}$$

- The DTFT of $x(n)$ is given by:

$$X(\omega) = \sum_{n=0}^{L-1} x(n)e^{-j\omega n} = \sum_{n=0}^{L-1} e^{-j\omega n} = \frac{\sin(\omega L/2)}{\sin(\omega/2)} e^{-j\omega(L-1)/2}$$

Thus,

$$X(\omega) = \frac{\sin(\omega L/2)}{\sin(\omega/2)} e^{-j\omega(L-1)/2}$$

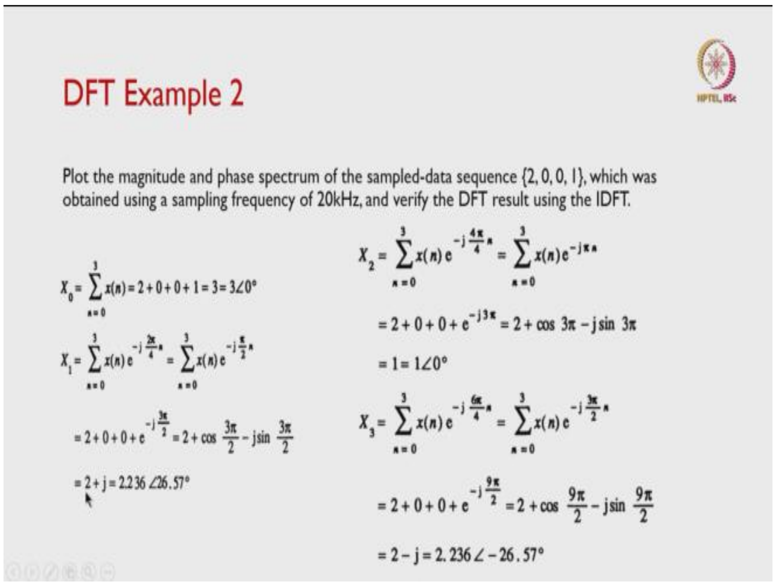
$$X(k) = \frac{\sin\left(\frac{2\pi k}{N}L/2\right)}{\sin\left(\frac{2\pi k}{N}/2\right)} e^{-j\frac{2\pi k}{N}(L-1)/2} = \frac{\sin(\pi kL/N)}{\sin(\pi k/N)} e^{-j\pi k(L-1)/N}$$

4 Rathna G N

So, now we will see with a few examples, so, we first will determine the endpoint DFT of the following sequence for $n \geq L$, what we call it. So, $x(n) = 1$ in this $0 \leq n \leq L-1$ and which is going to be 0 otherwise, so, then we can represent our DFT of $x(n)$ up to length $L-1$. So, that is $X(\omega)$ or $X(k)$ what we can take it, which is going to be, in this case we are assuming $X(\omega)$ that is DTFT discrete time Fourier transform, here, the amplitudes are discrete, and then our frequency is continuous. That is the reason why omega is there, which ω is going to vary 0 to 2π basically, then DTFT is given as $x(n)e^{-j\omega n}$. And then when we substitute that we have got $x(n)$ is equal to all one's. So, then what happens to that $\sum_{n=0}^{L-1} e^{-j\omega n}$, so, this what will be represented with $\frac{sin(\omega L/2)}{sin(\omega/2)} e^{-j\omega(L-1)/2}$.

So, what happens to our $X(\omega)$? So, this is what our equation is, I think something it must be boggling in your mind basically, we call this as a sine function. So, $X(k)$, if we calculate that in the DFT domain, this was in the DTFT. So, then what happens to this equation $sin\left(\frac{2\pi k}{N}L/2\right)$ and then here also $sin\left(\frac{2\pi k}{N}/2\right)$ and then there is $e^{-j\frac{2\pi k}{N}(L-1)/2}$. So, this is what, what we will be getting it if our $x(n)$ is we call it as a unity step function basically here you are for N basically n is, in this case L - 1 length.

(Refer Slide Time: 04:54)


DFT Example 2

Plot the magnitude and phase spectrum of the sampled-data sequence {2, 0, 0, 1}, which was obtained using a sampling frequency of 20kHz, and verify the DFT result using the IDFT.

$$X_0 = \sum_{n=0}^{3} x(n) = 2+0+0+1 = 3 = 3\angle 0^\circ$$

$$X_1 = \sum_{n=0}^{3} x(n) e^{-j\frac{2\pi}{4}n} = \sum_{n=0}^{3} x(n) e^{-j\frac{\pi}{2}n}$$

$$= 2+0+0+e^{-j\frac{3\pi}{2}} = 2+\cos\frac{3\pi}{2} - j\sin\frac{3\pi}{2}$$

$$= 2+j = 2.236\angle 26.57^\circ$$

$$X_2 = \sum_{n=0}^{3} x(n) e^{-j\frac{4\pi}{4}n} = \sum_{n=0}^{3} x(n) e^{-j\pi n}$$

$$= 2+0+0+e^{-j3\pi} = 2+\cos 3\pi - j\sin 3\pi$$

$$= 1 = 1\angle 0^\circ$$

$$X_3 = \sum_{n=0}^{3} x(n) e^{-j\frac{6\pi}{4}n} = \sum_{n=0}^{3} x(n) e^{-j\frac{3\pi}{2}n}$$

$$= 2+0+0+e^{-j\frac{9\pi}{2}} = 2+\cos\frac{9\pi}{2} - j\sin\frac{9\pi}{2}$$

$$= 2-j = 2.236\angle -26.57^\circ$$

So, then we will take a simple example to see that how we are going to compute our DFT from a small example. So, what it says is we have to do the plot and now calculate the magnitude plot the magnitude and phase spectrum of the sample data sequence it is just 2, 0, 0, 1 4 point of what we have to calculate, so, which was obtained using a sampling frequency of 20 kilohertz, and verify the DFT result using IDFT also whatever DFT values, we have got it, how to verify it, so, we have to do IDFT, then we have to get back the signal.

So, we will see with that, because it is the small value what we have taken the thing 4 bit, so, our X_0 what we will calculate first, so, for that $\sum_{n=0}^{3} x(n)$ because our $k = 0$ in this case, correct? So, e^{-j} will be one so, it is $x(n)$. So, it is nothing but 2 + 0 + 0 + 1 which is equal to 3, which is $|3|\angle 0°$ and then we will calculate X of X_1 basically, $\sum_{n=0}^{3} x(n)e^{-j}$.

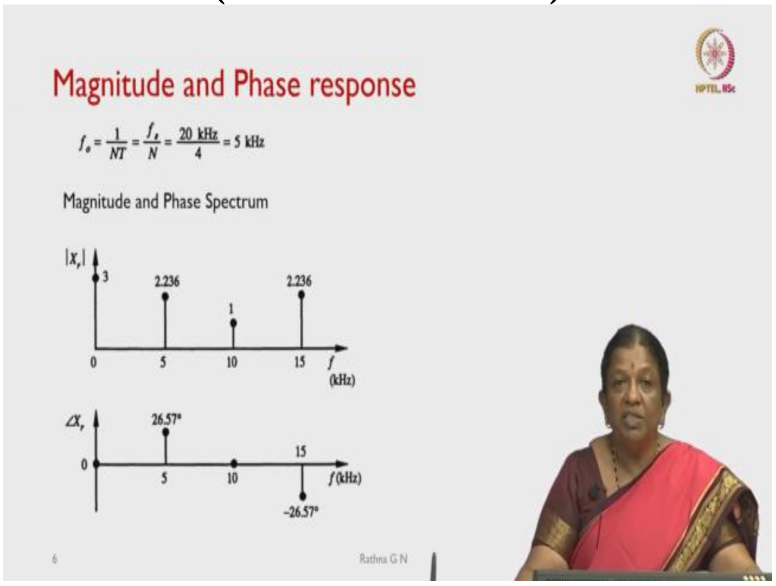
So, which is equal to what is it $x(n)e^{-j\frac{\pi}{2}}$ because it is n capital N = 4 so, which comes down to $-j\frac{\pi}{2}n$ basically. So, when you substitute the values for expand your summation it is going to be a first one is X_0 is 2. So, it is going to be 2 and the other 2 values what we have taken for simplicity is 0 and then the last one is when n = 3, so, this becomes $e^{-j\frac{3\pi}{2}}$.

So, if you substitute for e^{-j} in cos and sine you will be seeing $2 + \cos\frac{3\pi}{2} - j\sin\frac{3\pi}{2}$. So, you will be getting this as $2 + j$. So, this is 0 and this is -1 so, it will be plus $2 + j$ so, when you calculate your magnitude it is going to be 2.236∠26.57°. So, magnitude is computed with the square root of real square plus imaginary square.

So, that is the value you will get it and then the angle will be tan inverse what you can see imaginary divided by your real. So, now calculate the X_2 next page. So, here also you will be seeing that 3 components are left out it is going to be $e^{-j3\pi}$, so, it is going to be $\cos 3\pi - j\sin 3\pi$. So, this comes out as you can see that it is going to be minus one and this j term is 0.

So, it becomes a $2-1=1\angle 0°$, there are 2 ways of computing your X_3. So, one is here what is shown is direct computation. So, what you arrive at is $2-j$, so, this is added 2.236 at an angle of minus $26.57°$, this is one method. The other one is you can use the periodicity property basically, and we know that X_3 is equivalent to X_1 conjugate. So, then what happens I will be taking 2 and $-j$. So, this is the way what you will be substituting.

(Refer Slide Time: 09:06)


Magnitude and Phase response

$f_o = \frac{1}{NT} = \frac{f_s}{N} = \frac{20\text{ kHz}}{4} = 5\text{ kHz}$

Magnitude and Phase Spectrum

6

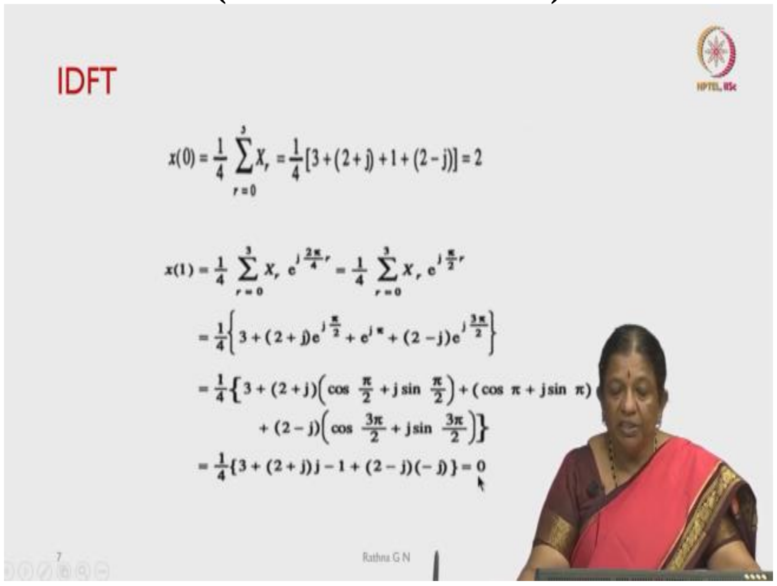
Rathna G N

So, now we will see how the magnitude and phase response is going to look like. So, we have been given the sampling frequency f_S is 20 kilohertz and then our $N=4$. So, if I substitute $\frac{f_S}{N}$ is nothing but 5 kilohertz, so, we will be seeing that f in the thing x axis will have points of 0, 5, 10 and then 15 are the sampling frequencies so, 4 points what we have to have our magnitude.

Magnitude of our X_r is going to be we know that first one is 3 what we have got it and second one is 2.236, 1 and then 2.236 and then we have calculated our angle also at these frequencies what you will be putting the $\angle$ so, first one is at 0 is 0 and here we have at $26.37°$ and then the at 10 kilohertz we have 0 and at 15 kilohertz it has minus $26.57°$, this is how manually you can do the thing.

So, we will seeing in the lab that how we can compute our magnitude and phase spectrum using either MATLAB or Code Composer Studio in using C language.

(Refer Slide Time: 10:30)



IDFT

$$x(0) = \frac{1}{4}\sum_{r=0}^{3} X_r = \frac{1}{4}[3+(2+j)+1+(2-j)] = 2$$

$$x(1) = \frac{1}{4}\sum_{r=0}^{3} X_r\, e^{j\frac{2\pi}{4}r} = \frac{1}{4}\sum_{r=0}^{3} X_r\, e^{j\frac{\pi}{2}r}$$
$$= \frac{1}{4}\left\{3+(2+j)e^{j\frac{\pi}{2}} + e^{j\pi} + (2-j)e^{j\frac{3\pi}{2}}\right\}$$
$$= \frac{1}{4}\left\{3+(2+j)\left(\cos\frac{\pi}{2} + j\sin\frac{\pi}{2}\right) + (\cos\pi + j\sin\pi) + (2-j)\left(\cos\frac{3\pi}{2} + j\sin\frac{3\pi}{2}\right)\right\}$$
$$= \frac{1}{4}\{3+(2+j)j - 1 + (2-j)(-j)\} = 0$$

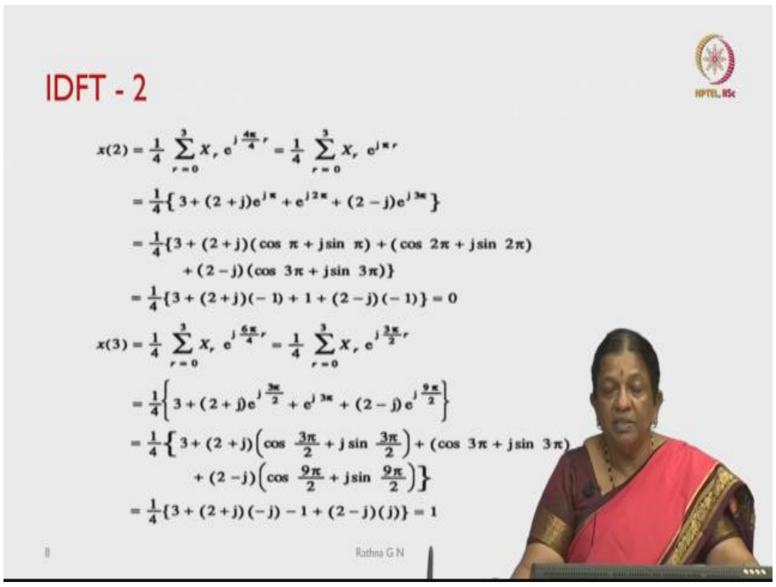
7 Rathna G N

So, now comes the thing so, we have to verify whatever DFT values we have got it using the IDFT that is the inverse discrete Fourier transform we are going to get back. So, we saw our analysis and synthesis equation. So, we will be making use of our synthesis equation, which is one by n is nothing but 1 by 4 in this case, so, our r will be varying between 0 to $3X_r$ what I had to put the thing, so, $\frac{1}{4}[3+(2+j)+1+(2-j)]0$.

So, you are already seeing that we have got back our original signal 2 that is first signal. So, this is going to be 3 + 1, 4 by 4 is sorry, 3 + 1 + 4 plus, what is the thing we have it here, you can see that this is 2 + 2 is 4. So, 4 + 4 is 8, what we have it 8 by 4 is nothing but $2+j$ and $-j$ are getting cancelled in this case. So, now, we will say $x(1)$ so, this is $\frac{1}{4}\sum_{r=0}^{3} X_r\, e^{j\frac{2\pi}{4}r}$.

So, which by expanding we are going to get it as that is we will be getting it $\frac{1}{4}$. So, we have to apply whatever DFT equation what we have got it values for them $3+(2+j)X_1$ and then this is $e^{j\frac{\pi}{2}} + e^{j\frac{\pi}{2}} + (2-j)e^{j\frac{3\pi}{2}}$ in this case, so, when we expand this, so, we are going to get it as 0. So, you can simplify this and you can see that the output is 0.

(Refer Slide Time: 12:39)



IDFT - 2

$$x(2) = \frac{1}{4}\sum_{r=0}^{3} X_r\, e^{j\frac{4\pi}{4}r} = \frac{1}{4}\sum_{r=0}^{3} X_r\, e^{j\pi r}$$

$$= \frac{1}{4}\{3 + (2+j)e^{j\pi} + e^{j2\pi} + (2-j)e^{j3\pi}\}$$

$$= \frac{1}{4}\{3 + (2+j)(\cos\pi + j\sin\pi) + (\cos 2\pi + j\sin 2\pi) + (2-j)(\cos 3\pi + j\sin 3\pi)\}$$

$$= \frac{1}{4}\{3 + (2+j)(-1) + 1 + (2-j)(-1)\} = 0$$

$$x(3) = \frac{1}{4}\sum_{r=0}^{3} X_r\, e^{j\frac{6\pi}{4}r} = \frac{1}{4}\sum_{r=0}^{3} X_r\, e^{j\frac{3\pi}{2}r}$$

$$= \frac{1}{4}\left\{3 + (2+j)e^{j\frac{3\pi}{2}} + e^{j3\pi} + (2-j)e^{j\frac{9\pi}{2}}\right\}$$

$$= \frac{1}{4}\left\{3 + (2+j)\left(\cos\frac{3\pi}{2} + j\sin\frac{3\pi}{2}\right) + (\cos 3\pi + j\sin 3\pi) + (2-j)\left(\cos\frac{9\pi}{2} + j\sin\frac{9\pi}{2}\right)\right\}$$

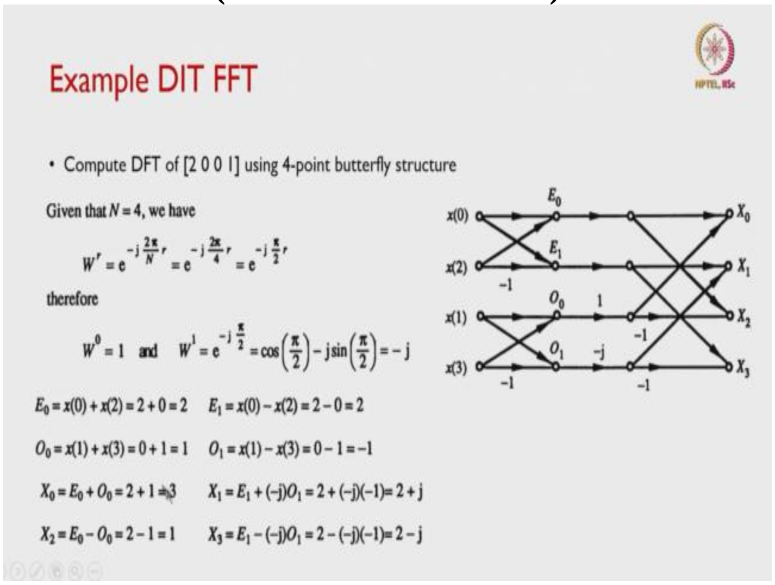
$$= \frac{1}{4}\{3 + (2+j)(-j) - 1 + (2-j)(j)\} = 1$$

8 Rathna G N

So, how about $x(2)$ now, same way what we have to work it out, substitute all the values and then you will be seeing that that also comes in 0. So, we know that using the symmetry property we can calculate $x(3)$ or manually by calculating to verify that you are going to get it 1 what you can see it, this case, we will not be able to compute the other value because it is 2, 0, 0, 1 is our value.

So, we had to go with manual computation and you will be seeing the last stage what you have it is these are the values by simplifying you will get it as 1. So, you will be seeing that what was the original case 2, 0, 0 here and 1 is the output. So, you will be seeing that DFT and then is analysis equation and IDFT is your synthesis. So, whatever value you have sent it you are getting it back basically so, no compression is happening.

(Refer Slide Time: 13:47)



Example DIT FFT

- Compute DFT of [2 0 0 1] using 4-point butterfly structure

Given that $N = 4$, we have

$$W^r = e^{-j\frac{2\pi}{N}r} = e^{-j\frac{2\pi}{4}r} = e^{-j\frac{\pi}{2}r}$$

therefore

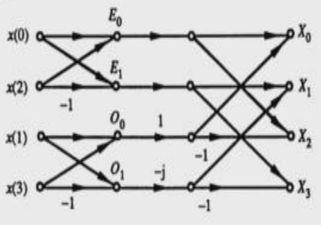
$$W^0 = 1 \quad \text{and} \quad W^1 = e^{-j\frac{\pi}{2}} = \cos\left(\frac{\pi}{2}\right) - j\sin\left(\frac{\pi}{2}\right) = -j$$

$E_0 = x(0) + x(2) = 2 + 0 = 2$ $\quad$ $E_1 = x(0) - x(2) = 2 - 0 = 2$

$O_0 = x(1) + x(3) = 0 + 1 = 1$ $\quad$ $O_1 = x(1) - x(3) = 0 - 1 = -1$

$X_0 = E_0 + O_0 = 2 + 1 = 3$ $\quad$ $X_1 = E_1 + (-j)O_1 = 2 + (-j)(-1) = 2 + j$

$X_2 = E_0 - O_0 = 2 - 1 = 1$ $\quad$ $X_3 = E_1 - (-j)O_1 = 2 - (-j)(-1) = 2 - j$



So, why I am calling it as compression sometimes we use our DFT for compression also we can eliminate some of the low values and use in compression. Now we will see that how we can use the previous one was decimation in time what we have used it using the FFT that is fast Fourier transform using the butterfly structure. So, we have $x(0)$ and next one is $x(2)$, $x(1)$ and $x(3)$ are the 4 inputs.

What we have to give it so this you will be seeing that this is $x(2) = 0$, and $x(1) = 0$, and then 1 is $x(3)$ so it does not make the order reverse in this case because both of them we have 0, then we have to calculate our this thing but twiddle factors since $n = 4$. So, our W^r what we call it or W^k which of the one we are can converse and we will be using notations it may little bit vary from 1 to 1, so, you can have 1 notation to use the thing.

So, it becomes $e^{-j\frac{2\pi}{N}r}$. So, which will be equivalent to, it is $e^{-j\frac{2\pi}{4}r}$. So, which is nothing but $e^{-j\frac{\pi}{2}r}$. So, we will see that W^0 always we know that it is 1 what is W^1 is nothing but putting $r = 1$ it is going to be $e^{-j\frac{\pi}{2}}$. So, which is nothing but $\cos\left(\frac{\pi}{2}\right) - j\sin\left(\frac{\pi}{2}\right)$. So, we know $\cos\left(\frac{\pi}{2}\right) = 0$ and $\sin\left(\frac{\pi}{2}\right) = 1$, so, we are going to end up with $-j$.

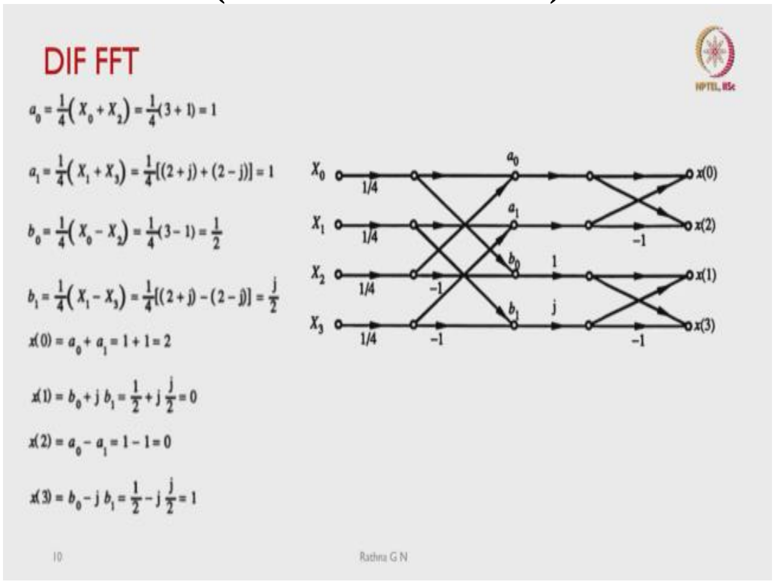
So, these are the only twiddle factors what we need in computation. So, you will be putting this is 1 and this is $-j$ here. So, now, do the simple multiplication and addition. So, what is my E_0 is nothing but $x(0) + x(2)$. Because we assume that this is 1 so, this is first one is $x(0) + x(2)$, which is nothing but 2 + 0 = 2. And then same way we will calculate E_1 here, which is nothing but $x(0) - x(2)$.

So, you can see that how we are going with the thing 2 - 0 = 2 and compute your O_0 and then O_1 this is an even part what it is and this is the odd part. So, what you will be getting 1 and then -1 in this case. So, these are the E_0 E_1 O_0 O_1, we have to calculate because it is in this thing in order output $x(0)$ $x(1)$ $x(2)$ $x(3)$ what we have it, so, what is $x(0)$? So, you will be seeing that $x(0)$ is $E_0 + rx(1)$.

So, it is 2 + 1 = 3 in this case and then our $x(1)$ is nothing but what is what we have E_1 and then next $-jO_1$. So, that is what, what it is written. So, it is nothing but E_1 what we have is $2 + -j$ into 1 is our thing is -1. So, this you will be getting it as $2 + j$ and you will be seeing the last one if you substitute that thing you will get it as $2 - j$ so, $x(3)$. So, whatever DFT you have seen that.

How difficult is to calculate here using the butterfly structure that is fast Fourier transform, we were able to do it in 2 steps to compute this values and 1 weight factor what we have to calculate and then put it in our butterfly structure.

(Refer Slide Time: 18:10)



DIF FFT

$a_0 = \frac{1}{4}(X_0 + X_2) = \frac{1}{4}(3 + 1) = 1$

$a_1 = \frac{1}{4}(X_1 + X_3) = \frac{1}{4}[(2 + j) + (2 - j)] = 1$

$b_0 = \frac{1}{4}(X_0 - X_2) = \frac{1}{4}(3 - 1) = \frac{1}{2}$

$b_1 = \frac{1}{4}(X_1 - X_3) = \frac{1}{4}[(2 + j) - (2 - j)] = \frac{j}{2}$

$x(0) = a_0 + a_1 = 1 + 1 = 2$

$x(1) = b_0 + j\, b_1 = \frac{1}{2} + j\,\frac{j}{2} = 0$

$x(2) = a_0 - a_1 = 1 - 1 = 0$

$x(3) = b_0 - j\, b_1 = \frac{1}{2} - j\,\frac{j}{2} = 1$

10 Rathna G N

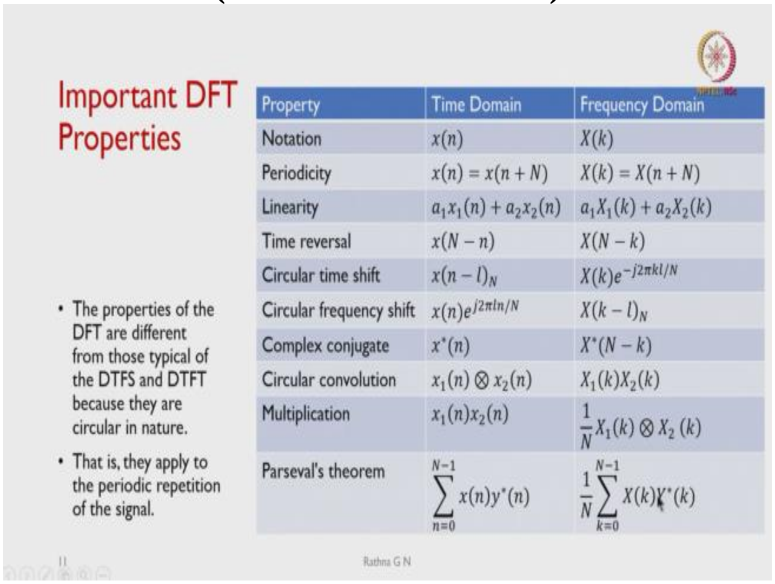
Now, we have to see that, whether we are going to get back the results using FFT. So, what is it here, $x(0)$ $x(1)$ $x(2)$ $x(3)$ are in sequence and our output will be out of order $x(0)$ that is a bit reversed output what we will be getting it $x(0)$ $x(1)$ $x(2)$ $x(3)$, this we call it as dissemination in frequency FFT. So, you will be seeing that we use this dissemination in frequency which is nothing but $\frac{1}{n}$ $\frac{1}{4}$what we have to put a scale all of them.

And then we have the butterfly or whatever we use for DFT which is going to be reversed in this case, as you can see the thing and you will be marking this as a_0 a_1 b_0 and b_1 and you are this thing, it is 1 coefficient here minus j conjugate you will be getting it as plus j in this case. So, at then rest of the butterfly, you will be multiplying with minus 1 at the second part of it, the top 1 is going to have plus 1 as you can see that thing here plus 1.

And here also plus 1. So, to this computation, you will be seeing that you have to get back your output. So, we have a_0 a_1 b_0 and b_1 first computed and in the next case, you will be calculating $x(0)$ $x(1)$ $x(2)$ $x(3)$ so, you are seeing that you are getting back to 2, 0, 0, 1 as your output.

This is how we are both of them decimation in time and decimation in frequency work for your analysis and synthesis equivalent of DFT.

(Refer Slide Time: 20:05)



Important DFT Properties

- The properties of the DFT are different from those typical of the DTFS and DTFT because they are circular in nature.
- That is, they apply to the periodic repetition of the signal.

Property	Time Domain	Frequency Domain
Notation	$x(n)$	$X(k)$
Periodicity	$x(n) = x(n+N)$	$X(k) = X(n+N)$
Linearity	$a_1x_1(n) + a_2x_2(n)$	$a_1X_1(k) + a_2X_2(k)$
Time reversal	$x(N-n)$	$X(N-k)$
Circular time shift	$x(n-l)_N$	$X(k)e^{-j2\pi kl/N}$
Circular frequency shift	$x(n)e^{j2\pi ln/N}$	$X(k-l)_N$
Complex conjugate	$x^*(n)$	$X^*(N-k)$
Circular convolution	$x_1(n) \otimes x_2(n)$	$X_1(k)X_2(k)$
Multiplication	$x_1(n)x_2(n)$	$\frac{1}{N}X_1(k) \otimes X_2(k)$
Parseval's theorem	$\sum_{n=0}^{N-1} x(n)y^*(n)$	$\frac{1}{N}\sum_{k=0}^{N-1} X(k)Y^*(k)$

11

Rathna G N

Now, some of the important DFT properties we are seeing in this slide. So, the properties of the DFT are different from those typical, what we use it for discrete time Fourier series and discrete time Fourier transform, because they are circular in nature. So, that is they applied to the periodic repetition of the signal here you can have DTFS or DTFT it can be this thing continuous signal minus infinity to infinity whereas, this uses a circular property.

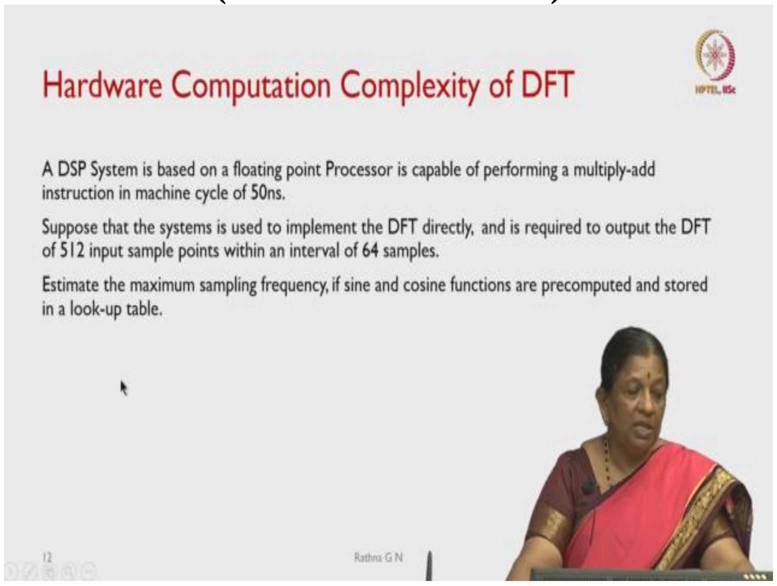
So, the first one is your notation, what is it time domain is represented as x of n and frequency domain we represent it as $X(k)$. So, the first one is the periodicity. So, $x(n) = x(n+N)$ is nothing but the state of what we will be getting it $x(n)$ itself. So, which is equivalent to $X(k) = X(n+N)$ in the frequency domain and what is our linearity property. So, if x_1 is multiplied by any coefficient a_1 and x_2 multiplied by a_2.

And if you add them the result in frequency domain also it should be pre multiplied by a_1 with respect to your frequency domain conversion that is $a_1X_1(k) + a_2X_2(k)$. So, you can have the time reversal in your time domain. So, here it is going to be in the frequency domain it is with respect to k what you are going to get the time reversal and then you can have this circular time shift with respect to n.

So, which is going to be reflected in your frequency domain as $X(k)e^{-j2\pi kl/N}$ and complex conjugate $x^*(n)$ is nothing but X^* of delayed function that is $(N-k)$. And if we do the circular convolution in the time domain, it is circular multiplication as you will be seeing $X_1(k)X_2(k)$. So, with respect to multiplication here, it becomes the convolution circular convolution divided by scaling factor $\frac{1}{N}$.

So, we know the Parseval's theorem. So, it is $x_1(n)$ into $y^*(n)$ in the time domain, so, which is represented as $\frac{1}{N}\sum_{k=0}^{N-1} X(k)Y^*(k)$.

(Refer Slide Time: 22:59)



Hardware Computation Complexity of DFT

A DSP System is based on a floating point Processor is capable of performing a multiply-add instruction in machine cycle of 50ns.

Suppose that the systems is used to implement the DFT directly, and is required to output the DFT of 512 input sample points within an interval of 64 samples.

Estimate the maximum sampling frequency, if sine and cosine functions are precomputed and stored in a look-up table.

12

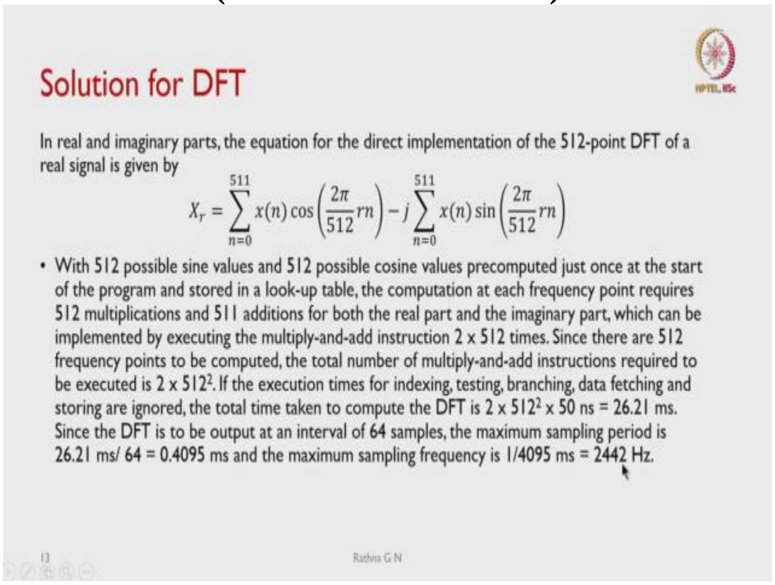
Rathna G N

So, now, we will see that how our, hardware complexity using DFT and FFT in processor is going to reflect. So, as an example, DSP system is based on a floating point processor we have considered and we say that it is capable of performing multiply and add instruction in 1 machine cycle, which is going to take 15 nano second. So, we have ignored in this case, all IO operations.

So, only we are looking into the computation aspects of our hardware. So, and we have next one what it is defined is that is suppose that the system is used to implement the DFT directly and is required to output the DFT of 512 input sample points within an interval of 64 samples, that is chunks of 64 samples are going to come out and then what we have to do it that is estimate the maximum sampling frequency if sine and cosine function are pre computed and stored in the lookup table.

So, we are not going to use the library function so, we are saving little time. So, we will be calculating for 512 points both sine and cos is computed and then kept in the table. So, we are using that to compute our DFT values.

(Refer Slide Time: 24:29)



Solution for DFT

In real and imaginary parts, the equation for the direct implementation of the 512-point DFT of a real signal is given by

$$X_r = \sum_{n=0}^{511} x(n)\cos\left(\frac{2\pi}{512}rn\right) - j\sum_{n=0}^{511} x(n)\sin\left(\frac{2\pi}{512}rn\right)$$

- With 512 possible sine values and 512 possible cosine values precomputed just once at the start of the program and stored in a look-up table, the computation at each frequency point requires 512 multiplications and 511 additions for both the real part and the imaginary part, which can be implemented by executing the multiply-and-add instruction 2 x 512 times. Since there are 512 frequency points to be computed, the total number of multiply-and-add instructions required to be executed is 2 x 512^2. If the execution times for indexing, testing, branching, data fetching and storing are ignored, the total time taken to compute the DFT is 2 x 512^2 x 50 ns = 26.21 ms. Since the DFT is to be output at an interval of 64 samples, the maximum sampling period is 26.21 ms/ 64 = 0.4095 ms and the maximum sampling frequency is 1/4095 ms = 2442 Hz.

13 Rathna G N

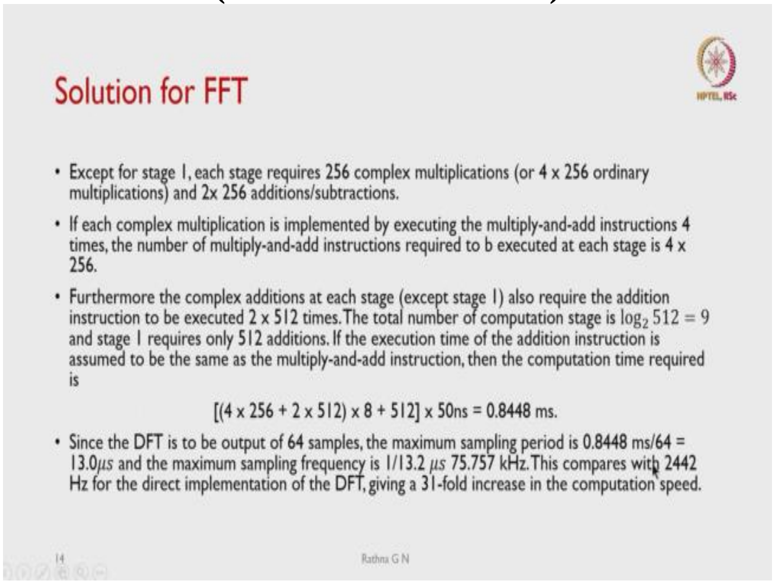
So, what is it? Using the DFT means, we know that are either $X_r = \sum_{n=0}^{511} x(n)\cos\left(\frac{2\pi}{512}rn\right) - j\sum_{n=0}^{511} x(n)\sin\left(\frac{2\pi}{512}rn\right)$ that is we have split out e - j into cos and sine function, so, we can see that this involves what is it? 512 possible sine values and 512 possible cosine values are pre computed.

So, we are not taking the computation time of it, because we have kept it in the lookup table and then what happens to this now, so, frequency point requires 512 multiplications what we need it here and then we need 511 additions as you think something it should be coming and n multiplications and n - 1 additions what we need it for real and then imaginary part separately what we have to do it then it is order of n if I take it 2 into 512 times what we want it, but what we have is k our r is going to go between 0 to 511.

So, then what happens to this it is 2 into 512 square is our computation time to do our endpoint DFT. So, and then we have been given we are ignoring because multiplication and addition we are combining it which is order of n square. So, which takes 50 nanosecond what we have been given with so, then this comes to about 26.21 millisecond. And we have been given that 64 samples chunks are going at a time.

So, we will divide this 26.21 by 64 which gives us 0.4095 milliseconds to do DFT for 512 length. And we know that maximum frequency what we can operate this circuit with this 1 by 4095 millisecond, which comes to about 2442 hertz or 2.442 kilo hertz so, this is what with the DFT.

(Refer Slide Time: 27:00)



Solution for FFT

- Except for stage 1, each stage requires 256 complex multiplications (or 4 x 256 ordinary multiplications) and 2x 256 additions/subtractions.
- If each complex multiplication is implemented by executing the multiply-and-add instructions 4 times, the number of multiply-and-add instructions required to b executed at each stage is 4 x 256.
- Furthermore the complex additions at each stage (except stage 1) also require the addition instruction to be executed 2 x 512 times. The total number of computation stage is $\log_2 512 = 9$ and stage 1 requires only 512 additions. If the execution time of the addition instruction is assumed to be the same as the multiply-and-add instruction, then the computation time required is

$$[(4 \times 256 + 2 \times 512) \times 8 + 512] \times 50\text{ns} = 0.8448 \text{ ms}.$$

- Since the DFT is to be output of 64 samples, the maximum sampling period is 0.8448 ms/64 = 13.0μs and the maximum sampling frequency is 1/13.2 μs 75.757 kHz. This compares with 2442 Hz for the direct implementation of the DFT, giving a 31-fold increase in the computation speed.

14 Rathna G N

So, we will see with the FFT how we are going to compute it, except for stage 1, which is going to have only additions. So, each stage we had to do n by 2 complex multiplications that is 256 complex multiplications and we have discussed in the last class that each complex multiplication is going to take 4 real or ordinary multiplication. So, it becomes 4 into 256 multiplications and we said addition is going to cause us 2 into 256 additional subtractions.

So, each complex addition is going to take 2 additions. So, that is the reason why 2 into 256 number of additions for the rest of the stages what we need it. So, if each complex multiplication is implemented by executing the multiply and add instructions, 4 times the number of multiply and add instructions required to execute at each stage that is 4 into 256 what we want, it has to be executed.

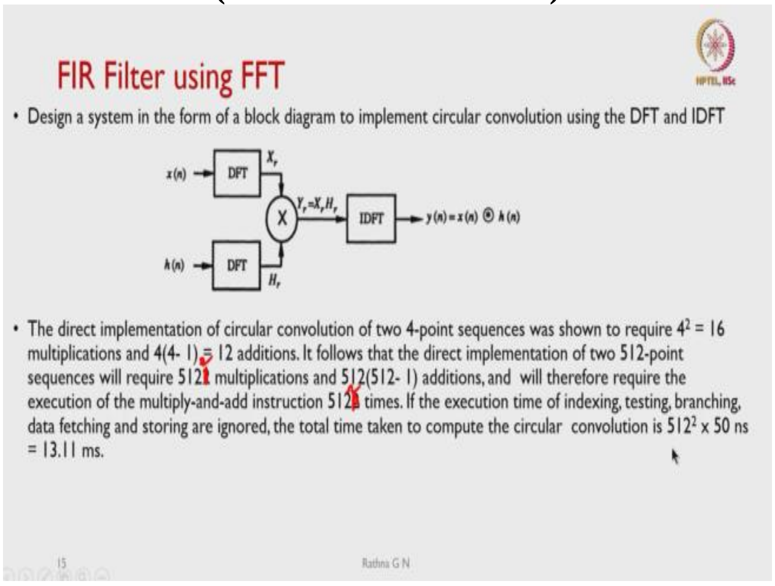
Further the complex additions at each stage except stage one also require an addition of 2 into 512 times we said additions. So, what happens is the total number of computation stages log 2 512 which is nothing but 9 stages what we have it and we set stage 1 we do not need any complex multiplication in the thing. So, we only need 5 additions in this case. So, if the execution time of addition instruction is assumed to be same as the multiply and add instruction.

So, although we know that multiply may take little more time, but we are assuming at present it is going to have addition also same time, then the computation time required for 512 point FFT what we are putting it so, this is 4 into 256 + 2 into 512, what we have it our additions as you can see the thing complex additions is nothing but 2 into 512 into because the first stage we are having only additions which is assumed here 512 stages.

So, n - 1 stages here 9 - 1 is going to be 8. So, these are the ones which are causing our complex multiplication and then complex addition which is separately taken into account. And then now whatever complex addition with multiplication is coming we are assuming that it is consumed in this, this is the other additions what we are showing it here. So, into 50 nanoseconds is going to give us 0.8448 millisecond.

And then we said that it is 64 samples in this time period what we are outputting it. So, divided by that will give us 13.0 microsecond and the maximum sampling frequency is going to be 1 by this, which is going to be 75.757 kilohertz, as you can see, 2 point some kilohertz and using DFT using FFT we are going to have 75.75 kilohertz so, that is the comparison. So, you will be seeing that it is equal to 31 fold increase in the computation speed.

(Refer Slide Time: 30:45)



FIR Filter using FFT

- Design a system in the form of a block diagram to implement circular convolution using the DFT and IDFT

$x(n)$
DFT
X_r
$h(n)$
DFT
H_r
X
$Y_r = X_r H_r$
IDFT
$y(n) = x(n) \circledast h(n)$

- The direct implementation of circular convolution of two 4-point sequences was shown to require $4^2 = 16$ multiplications and $4(4-1) = 12$ additions. It follows that the direct implementation of two 512-point sequences will require 512^2 multiplications and $512(512-1)$ additions, and will therefore require the execution of the multiply-and-add instruction 512^2 times. If the execution time of indexing, testing, branching, data fetching and storing are ignored, the total time taken to compute the circular convolution is $512^2 \times 50$ ns = 13.11 ms.

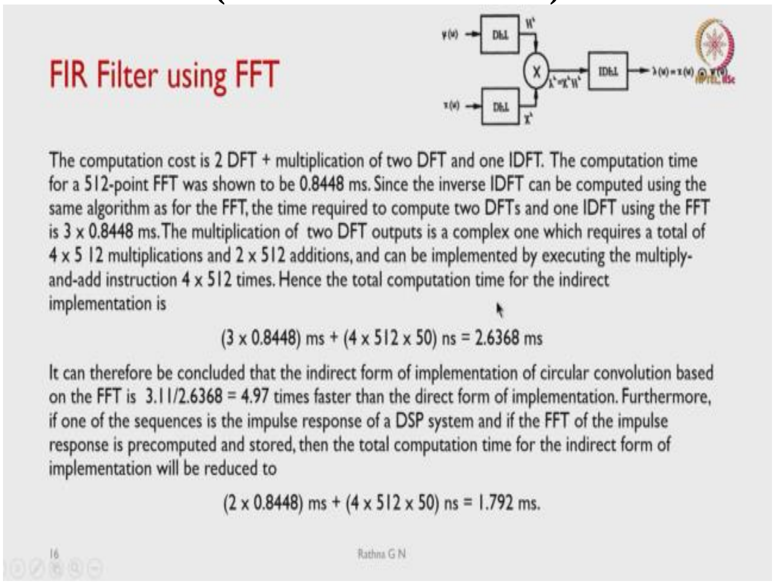
15 Rathna G N

So, now, we will see that how we can implement FIR filter using FFT. So, we will be taking the DFT of our input signal and DFT have our coefficient and then we have to do complex multiplication here $X_r H_r$ and then we are going to take IDFT so, this is what our circular convolution which is going to happen. So, how we are going to calculate this, if we are taking a 4 point.

Then we know that 4 square is 16 what we are going to get the thing and then multiplication and 4 into 4 - 1 12 additions, why in the direct circular convolution for 4 point now, we are assuming the same thing and then now, trying to see that, whether we are going to have 512 multiplications. So, here will require 512 multiplications and 512 into 512 - 1 whatever we said that additions for using DFT.

So, we will be seeing that add instruction we can do it as 512 square basically what we are going to have it as it sees 512 square here it is 512 square. So, what we say is 512 square into 50 nanoseconds, so it is going to be 13.11 millisecond what we need it.

(Refer Slide Time: 32:15)


DFT
DFT
X
IDFT

FIR Filter using FFT

The computation cost is 2 DFT + multiplication of two DFT and one IDFT. The computation time for a 512-point FFT was shown to be 0.8448 ms. Since the inverse IDFT can be computed using the same algorithm as for the FFT, the time required to compute two DFTs and one IDFT using the FFT is 3 x 0.8448 ms. The multiplication of two DFT outputs is a complex one which requires a total of 4 x 5 12 multiplications and 2 x 512 additions, and can be implemented by executing the multiply-and-add instruction 4 x 512 times. Hence the total computation time for the indirect implementation is

(3 x 0.8448) ms + (4 x 512 x 50) ns = 2.6368 ms

It can therefore be concluded that the indirect form of implementation of circular convolution based on the FFT is 3.11/2.6368 = 4.97 times faster than the direct form of implementation. Furthermore, if one of the sequences is the impulse response of a DSP system and if the FFT of the impulse response is precomputed and stored, then the total computation time for the indirect form of implementation will be reduced to

(2 x 0.8448) ms + (4 x 512 x 50) ns = 1.792 ms.

16

Rathna G N

So, now we will see using FFT, although I am going to take it DFT and this thing, so, previously we calculated each FFT needs 0.8448 millisecond and then how many times I have to do it here 1, 2, and then 3 of them what I have to do it because even IDFT is equivalent to DFT. So, we can now compute it as 3 into 0.8448 millisecond plus we have to do this complex multiplication here.

So, each complex multiplication as we know we have it as 4 normal multiplication 4 into 512 into 50 so, this is the additional. So, which runs at 2.6368 millisecond and then we are calculating that is 3.11 milliseconds in the thing here it is 2.6368 millisecond so we are still using all this complex method our FFT method is still 4.97 faster than that direct DFT and one more thing one can keep in mind that.

Because this is a pre define coefficient basically so I will not be recomputing every time only our input signal which is coming continuously so if we apply for DFT and keep it pre computed values here so we can avoid one of the DFT online so then it becomes this one 2 into 0.8448 so which reduce the so 1.792 millisecond compared to our 2.6368 millisecond so still we will be gaining the thing.

(Refer Slide Time: 34:08)

M2U18

- Overlap Add Method

17 Rathna G N

So, we will see in the next class the next class because we say that our $x(n)$ is continuous how we will use the overlap add or save method to compute our FFT. Thank you.