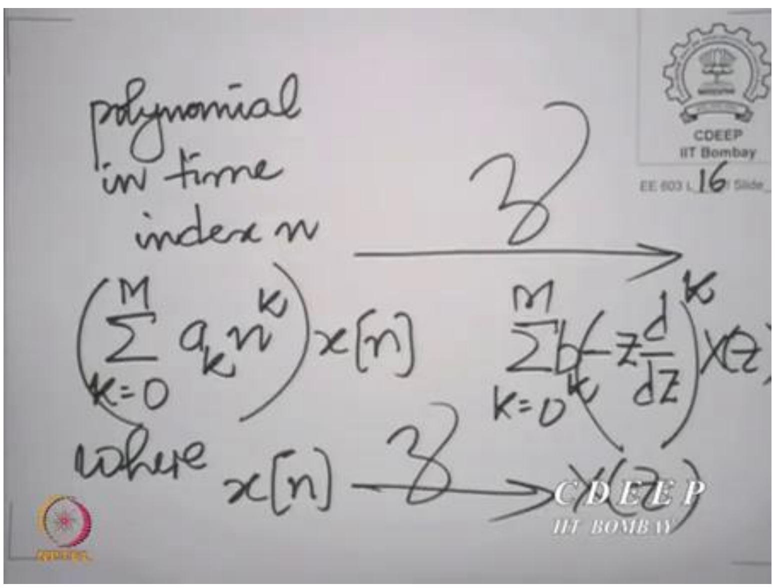
Digital Signal Processing & Its Application
Professor Vikram M. Gadre
Department of Electrical Engineering
Indian Institute of Technology, Bombay
Lecture 16a
Recap + Why are Rational Systems so Important?

A warm welcome to the sixteenth lecture on the subject of Digital Signal Processing and its Applications. We will take a minute to recapitulate what we did in the previous lecture, we had been looking at the rational Z transform in the previous lecture and we had evolved different ways to deal with the inversion of the rational Z transforms. Eventually, we had looked at partial fraction expansion and the use of the derivative; I mean the derivative of the Z transform.

So, we multiply the sequence by n when you take the derivative, eventually there is an association between multiplication of the sequence by the time index and taking the derivative in the Z domain. Of course, you know, you do need to multiply by the factor minus Z after taking the derivative and so on. But there is a relationship between multiplying by n or the time index or a polynomial in the time index. So, polynomial in the time, so let us write this down.

(Refer Slide Time: 01:23)



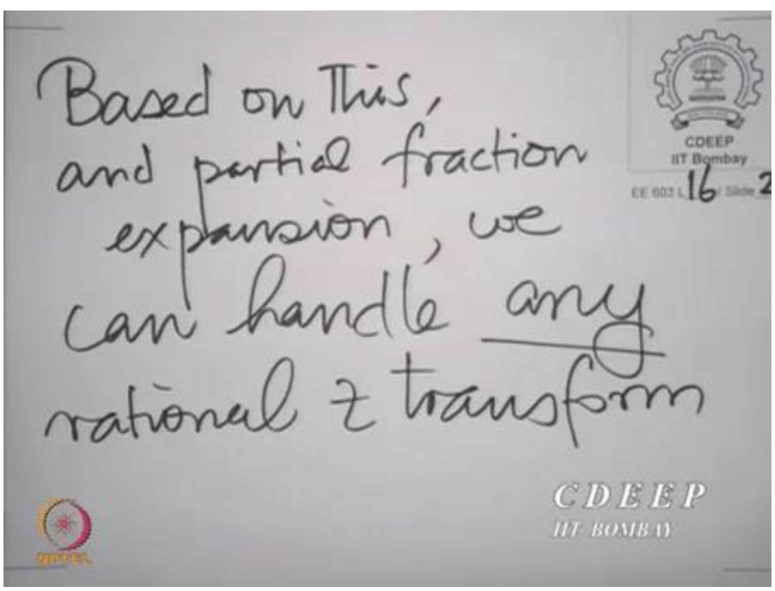
You see polynomial in the time index, polynomial in time index n. Eventually something of the form summation k going from 0 to say M, ak n raise to the power k multiplied by x n involves taking Eventually a linear combination of the kth derivative, this corresponds to summation going from 0 to M. Now, please note, we should interpret this carefully, what I am saying here minus Z d dZ to the power k XZ.

This means applying the operator, first take the derivative with respect to Z and then multiply by minus Z, together this should be treated as an operator. And, for example, if you are doing this twice, if k is equal to 2, what you mean is take the derivative with respect to Z multiply by minus Z, then take the derivative again with respect to Z and multiply by minus.

So, apply the whole operator minus said Z, d dZ twice that is how 1 should interpret it. So, this, this multiplied by bk, some bk, some constants. So Eventually, there is a, there is a one to one association in the Z domain between a polynomial in the time index multiplying the sequence and a polynomial in this operator minus Z d dZ multiplying the transform, operating on the Z transform.

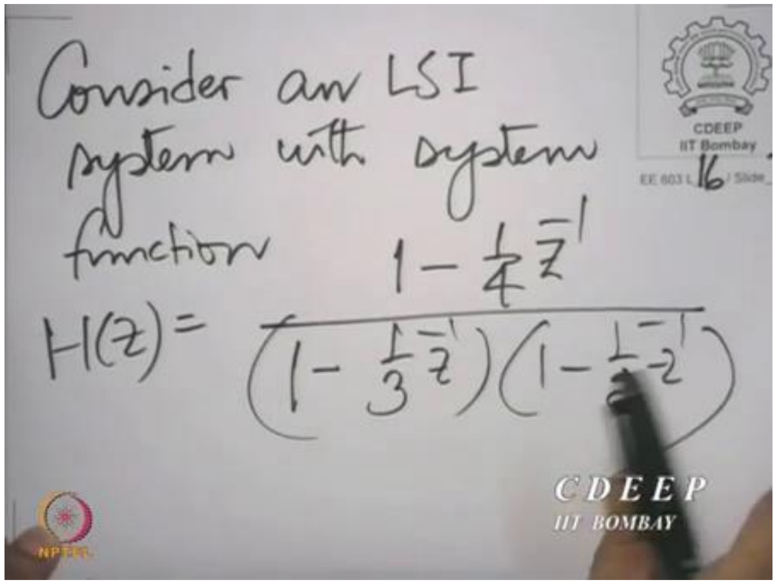
So, I also asked you to put down this relationship precisely by using inductive reasoning, I have left it to you as an exercise to do that. And based on this principle, we could see that, we could handle any Z transform, any rational Z transform, based on this principle and partial fraction expansion.

(Refer Slide Time: 04:10)



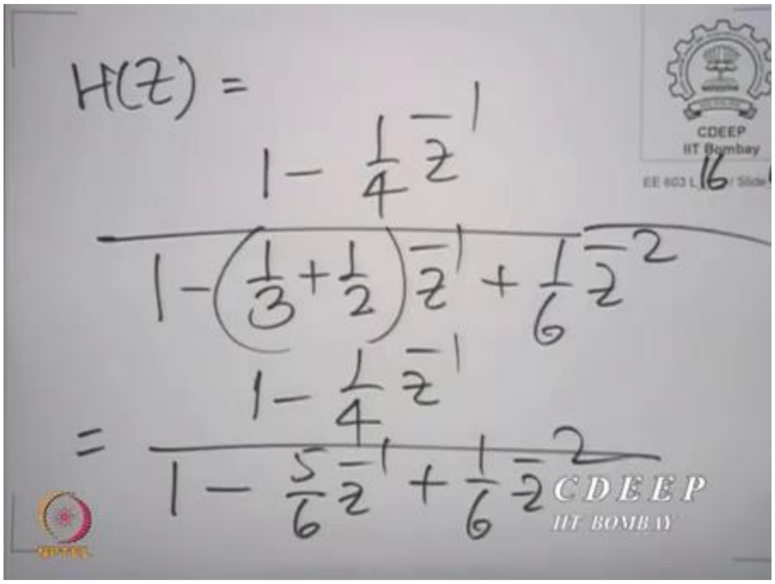
We can handle any rational Z transform. Now, one observation that we made last time was that you know, the rational Z transform is kind of frequent in occurrence that is why we said that we were spending so much of time on the rational Z transform. We shall take the next few minutes to understand why this is the case. Why is it that the rational Z transform is so important? Why does it, why does it, you know, mean a lot to us. In fact, let us assume that we have a system described by a system function which is rational.

(Refer Slide Time: 05:32)



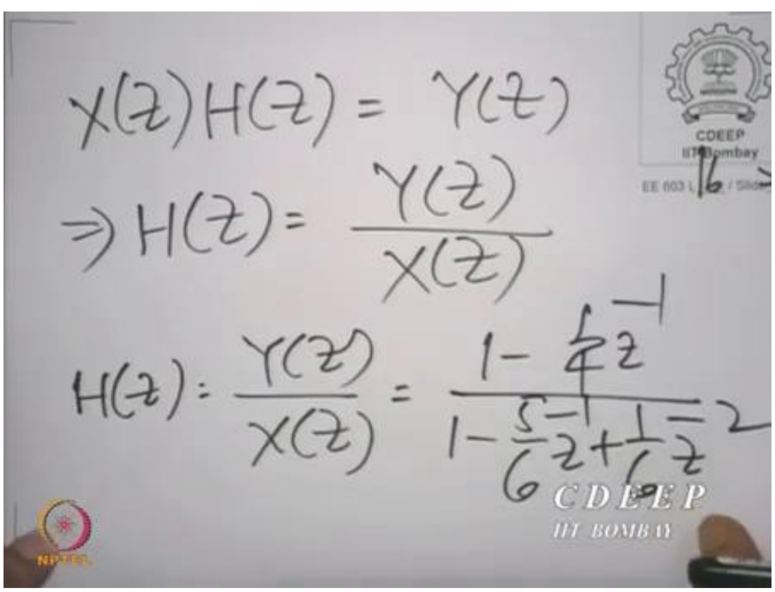
So, for example, let us take a concrete system as an example, where the transfer function or the system function is rational. Consider an LSI system with system function, let us say, 1 minus one fourth Z inverse divided by 1 minus 1 third Z inverse into 1 minus half Z inverse. This is capital H of Z; it is eventually the Z transform of the impulse response of that system. Now let us expand, let us draw out H Z.

(Refer Slide Time: 06:22)



H Z can be rewrite as 1 minus one fourth Z inverse divided by 1 minus one third plus half Z inverse plus one sixth Z raise to power minus 2 which of course becomes 1 minus one forth Z inverse 1 minus 3 plus 2 makes it 5 by 6 Z inverse plus 1 by 6 Z raise to power minus 2. Now what is H Z, H Z is the ratio of the output Z transform to the input Z transform. In fact, that is obvious because input Z transform multiplied by X Z would give you the output Z transform.

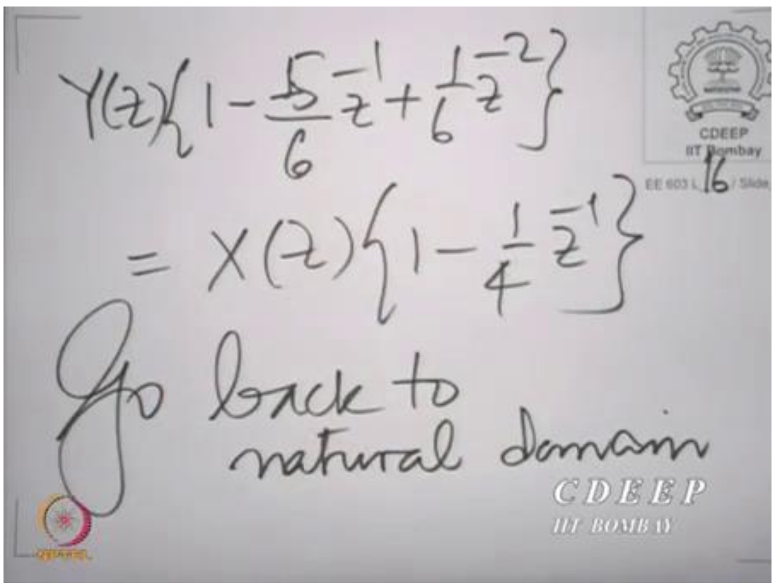
(Refer Slide Time: 07:11)



X Z multiplied by H Z would give you Y Z. And therefore, another way to understand the system function is the ratio of the output Z transform to the input Z transform. This also tells us something about LSI systems, if you look at the ratio of the output Z transform to the input Z transform. And if you find that this ratio is independent of the input, then the system is LSI. That is another way of looking at it.

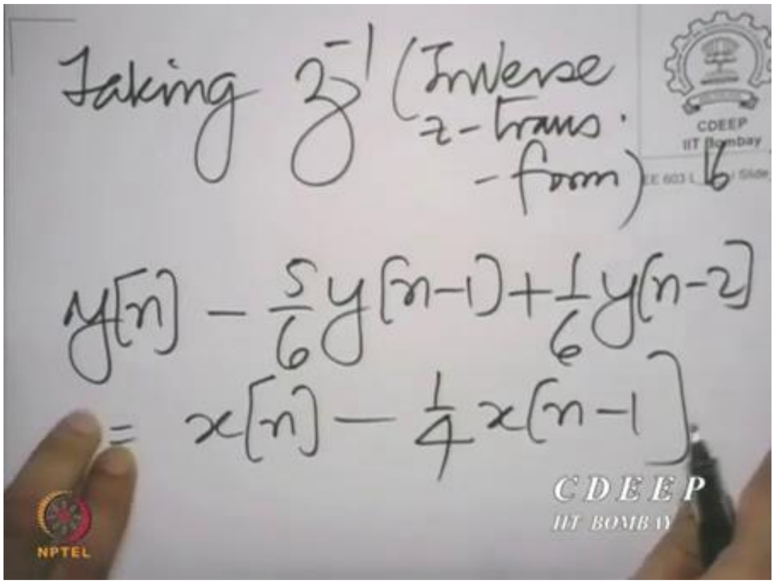
In the same domain, the fact that the output set transform divided by the input Z transform is independent of the input makes the system LSI. So in effect what we are saying is that, H Z which is Y Z by X Z in this case, is 1 minus one fourth Z inverse by 1 minus 5 by 6 Z inverse plus 1 by 6 Z raise to the power minus 2. And we can of course cross multiply.

(Refer Slide Time: 08:24)



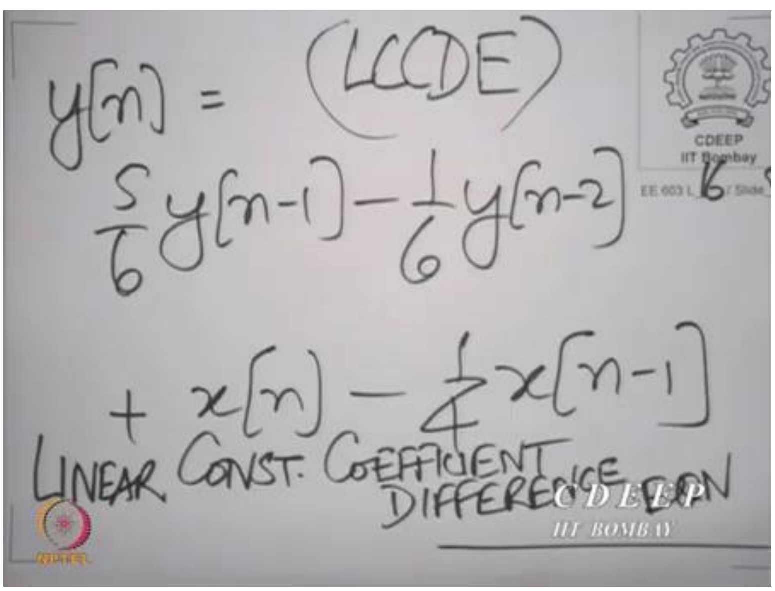
We can cross multiply to get Y Z into 1 minus 5 by 6 Z inverse plus 1 by 6 Z raise to the power of minus 2 is X Z into 1 minus one fourth Z inverse. And we can go back to time, go back to the natural domain, by taking the inverse Z transform both sides. So of course, it is very easy to see that you would get y n here and here, you would get minus 5 by 6 the inverse Z transforms Z inverse Y Z is Y of n minus 1, eventually y delayed 1 and so on

(Refer Slide Time: 09:08)



So, one can take the inverse Z transform on both sides. We will denote this as the inverse Z transform. You have y n minus 5 by 6 y of n minus 1 plus 1 by 6 y of n minus 2 is xn minus 1 by 4 x n minus 1. And of course, we can rewrite this, so you have, you know what, what we want to do ultimately is to write y n in terms of past values of y and x and its past values. So, we can keep y n on one side and all the other terms to the other side.

(Refer Slide Time: 09:08)



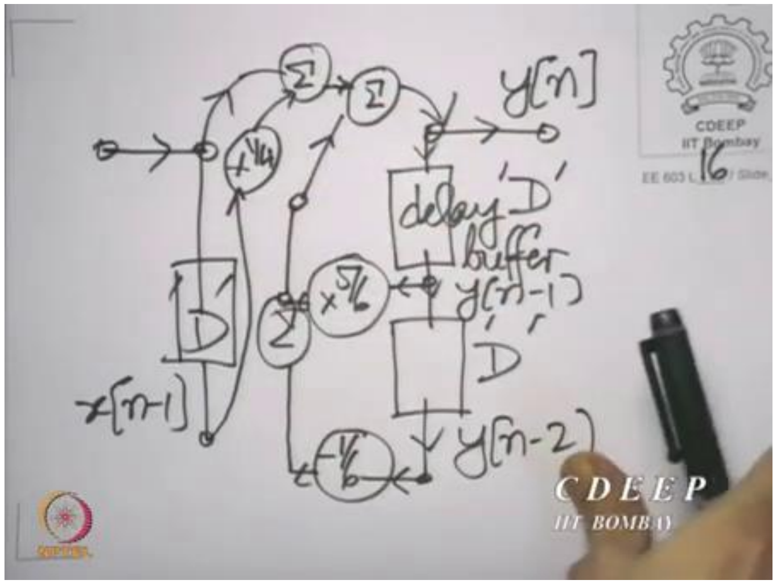
So, we have y n is 5 by 6 y n minus 1 minus 1 by 6 y n minus 2 plus x n minus 1 by 4 x n minus 1. And this is an example of what is called a Linear Constant Coefficient Difference Equation, it is abbreviated by LCCDE. Difference equation; let us understand the terms one by one. Difference equation is a relation between the input and the output in the sample natural domain. The difference equation is a relation between the discrete input and the discrete output.

A constant coefficient difference equation is a relationship where the constants that are involved were the multipliers that are involved are all constant. So, here these are the multipliers, 1 minus one forth 5 by 6 and minus 1 by 6 all of them are constant, that is why it is a constant coefficient difference equations. And linear, linear, because it obeys this entire input output relationship obeys the principle of superposition.

That means it obeys additivity in homogeneous and therefore, this is called a linear constant coefficient difference equation. If you drop the terms one by one, for example, you could have a constant coefficient difference equation which is not linear. For example, suppose in this equation you brought in y of n minus 2 the whole squared then it would become a constant coefficient difference equation but not linear and so on.

And there could be of course, a linear equation but not constant coefficient. For example, some of these terms could depend on n here. So, each of the, each of these terms in the expression LCCD is important. Now, LCCDE is a very important you see these, this kind of an equation is very fundamental to realizability. In fact, let us spend a minute on seeing how you can realize this very relationship here. That is, how would you, how would you translate this into a hardware structure or a software structure, we can see this right away.

(Refer Slide Time: 13:14)



So, in fact, suppose you happen to have, you know, a circuit which constantly generates y n, that means you have, you know let us assume that you have a timing mechanism, which times the samples, the sampling points and the samples. You know, in every in every unit time, one sample of the input is fed, and one sample of the output is generated. So, you have a timing mechanism, you have to have a timing mechanism.

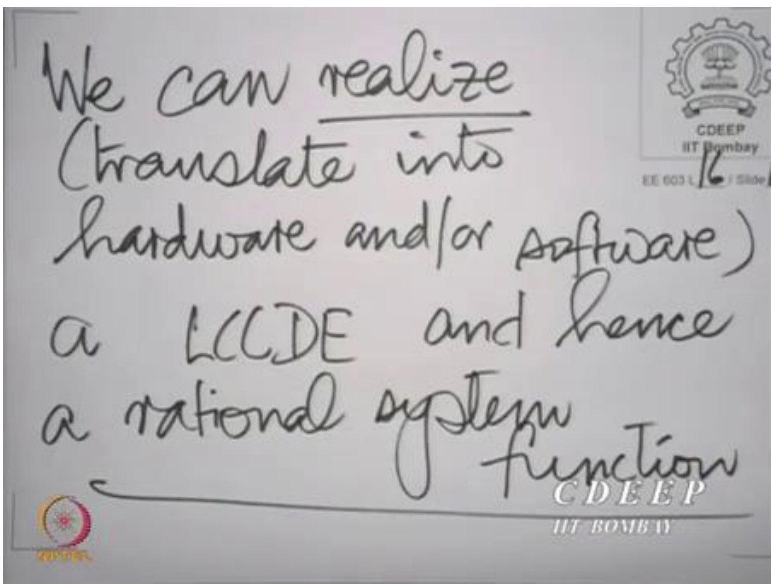
Now, in that timing mechanism, what you do is introduce what you call memory, so you introduce a delay buffer, which we will denote by D. In fact, introduce two of them. So, when you feed y n to a delay buffer, what you get is y of n minus 1, that means a delay buffer stores the previous value of the output, I mean stores the previous value of the input given to it. So, in this case, it is y n.

But in general, a delay buffer stores the previous value of the input given to it. And of course, here you would get y of n minus 2. And similarly, you could have a delay buffer operating on the input and that would give us x of n minus 1. Now, what does this equation tell me, this equation tells me that I have taken y of n minus 1 multiplied it by 5 by 6, I have taken y of n minus 2 and multiplied it by minus 1 by 6, I have added these 2 and show edition by a summation, here, I have added these two.

I have also taken the input and added to it minus 1 by 4 times the input delayed by one sound. And I have added these two together and thus generated the output. I have taken care here to allow only for what are called two input adders. So, we will assume that all the elements need to be of uniform structure. So, all the ads are off to inputs, of course, you could have additions with more than two inputs, because addition is a commutative and associative operation.

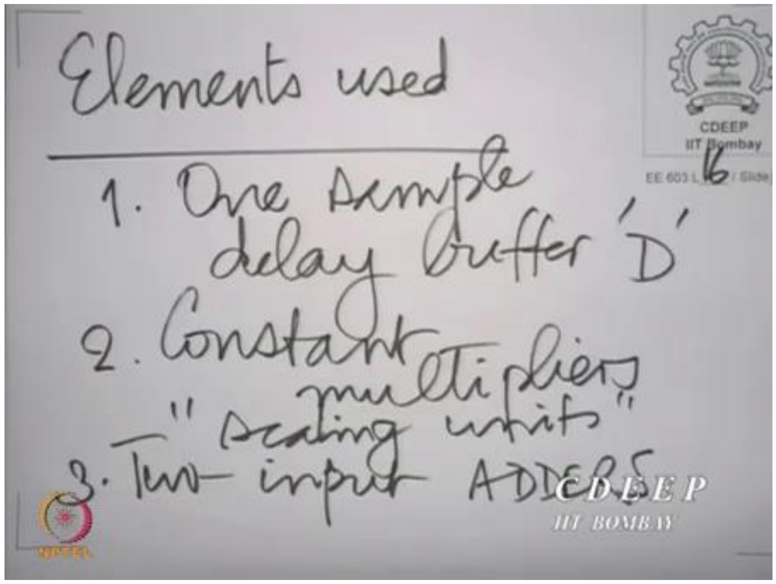
You can add more than two at a time, but we will assume for the sake of modularity and implementability that you use a two input add. What we have done is to realize this relationship this linear constant coefficient difference equation by using three kinds of elements; delay buffers, these Ds, multipliers, constant multipliers and two input adds. Let us write that down.

(Refer Slide Time: 16:40)



We can realize, realize means translate into hardware or and or software, a linear constant coefficient difference equation. And hence rational system function, rational system function is realizable. It is realizable because you can use a finite amount of hardware and software to translate it into an implementable structure, finite is important. What kind of hardware and software have we used here; let us again list that down.

(Refer Slide Time: 17:53)



Elements that are used, one sample delay buffer which we denoted by D, constant multipliers or scaling units, two input adders, these are three kinds of elements that we use. Of course, you can very easily see that each of these elements can either be thought of as a hardware element or as a software element. For example, if you think about the one sample delay, you could think of it as a little program statement which stores the previous value and keeps it at a memory location.

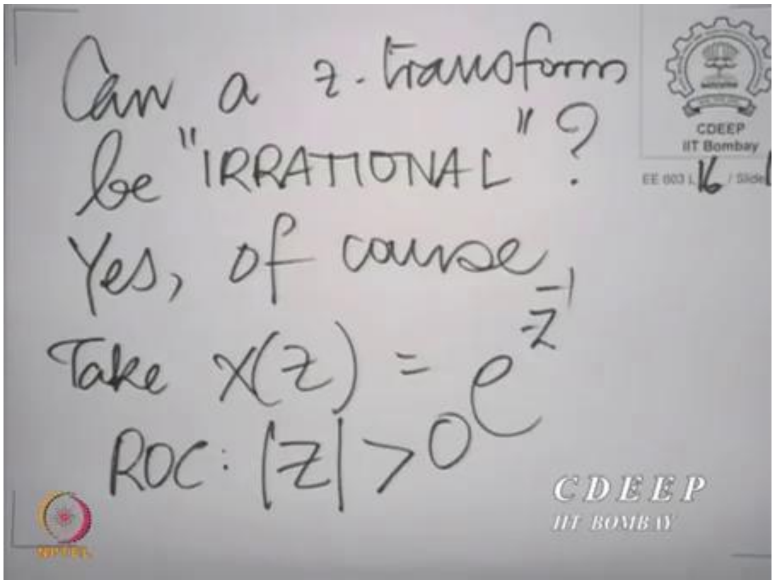
And of course, in hardware you can think of it as a register or a combination of registers. Similarly, a constant multiplier can be thought of either as a hardware element or as a software element. In software, it is a one line, multiplication by a constant. In hardware eventually, a multiplier implemented with any of the known multiplier algorithms, constant multiplier of course.

And you know that there are very efficient ways of implementing multipliers, if you know them to be constant multiplier. For example, one could use distributed arithmetic or any other efficient way of doing multiplication both algorithms. Anyway, so much so for the fact that and two input adders again, you know how to do that with two's complement or any other structure, so we know how to implement it in hardware and of course in software again it is a one line statement.

So, all of these are clearly unit statements in software or unit elements in hardware, modular elements in hardware. And with a combination of them we can realize a rational system function. So rational system function is realizable, now, to contrast let us take an example of an irrational.

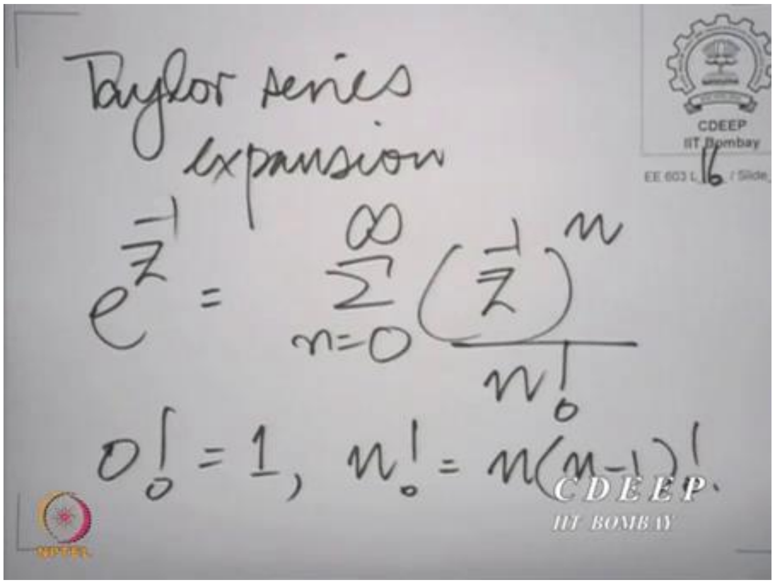
You see all this while we have been saying rational, so the natural question that comes to mind is, can you have an irrational system function?

(Refer Slide Time: 20:33)



So, can a system function or can a Z transform be irrational, and the answer of course has to be yes, of course. Take X Z is e raise to the powers Z inverse with the region of convergence, mod Z greater than 0. Now, note that here, you cannot include the point Z equal to 0 in the region of convergence. In fact, this is an example of what is called an essential singularity. You know, in some sense a singularity of infinite degree, we have seen in a minute.

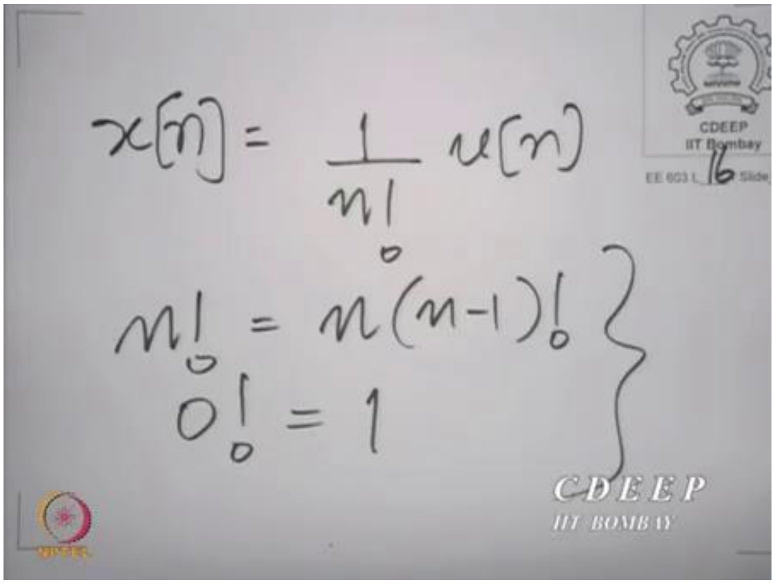
(Refer Slide Time: 21:38)



Now, e raise to the power Z inverse is very easy to expand, you know, those of us, all of us could have been exposed to the Taylor series at some, so the Taylor series expansion of e raise to the powers Z inverse, Eventually summation n going from 0 to infinity, Z inverse, raise to the power of n divided by n factorial, with 0 factorial defined as 1 and n factorial defined as n into n minus 1 factorial.

This is well known; this is eventually the Taylor series expansion of e raise to the power of x. Now when we write down the Taylor series expansion, it is very easy to identify what sequence corresponds to this X Z that is very easy.

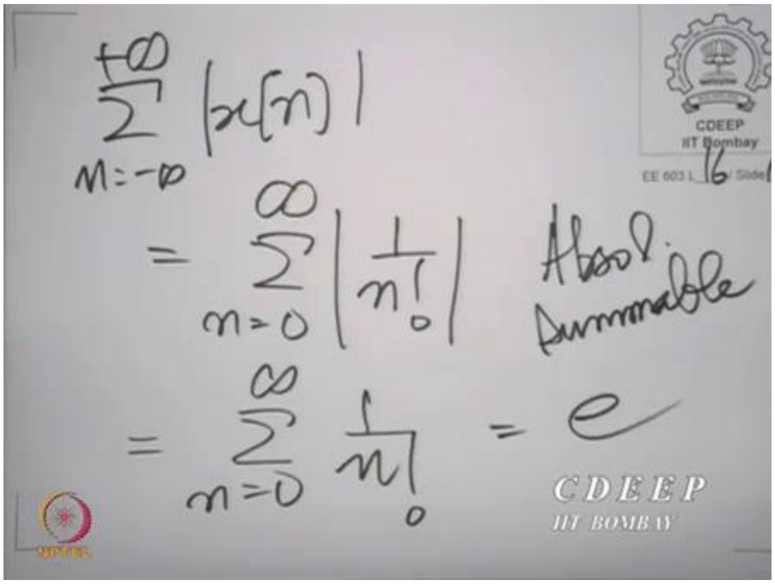
(Refer Slide Time: 22:33)



In fact, it turns out that this sequence x n would simply be 1 by n factorial u n, where n factorial has been defined already and u n, u n is the unit step. In fact, very interestingly, if an LSI system were to have this impulse response, suppose, x n where the impulse response of an LSI system, we could ask whether the system is causal and stable.

And it is very easy to see the system is of course, causal because the impulse response is 0 for n less than 0. So, of course, if this were to be the impulse response for LSI system, then LSI system would be causal. Would it be stable, how would we answer the question, we would need, need to look at the absolute sum of the impulse response. And in fact, the absolute sum of x n is very easy to calculate.

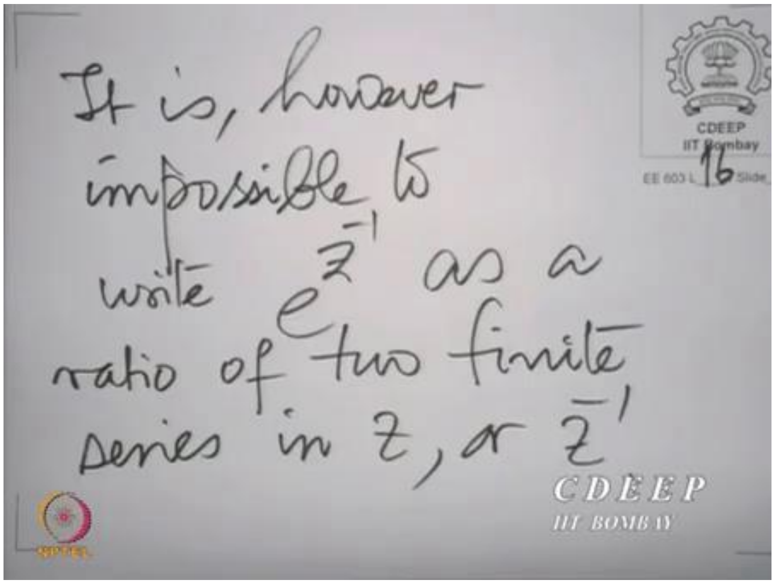
(Refer Slide Time: 23:50)



So, summation n over all integers mod x n is very easy to calculate this eventually summation n going from 0 to infinity mod 1 by n factorial, which is eventually 1 by n factorial, because n factorial is positive, and this is very easy to evaluate. This eventually e raise to the power of x evaluated at x equal to 1, the standard, neparian base and that is of course finite.

And therefore, this is absolutely summable. And therefore, the system is stable. So, if a system were to have this impulse response, it would be both causal and stable. However, e raise to the power Z inverse can never be written as a ratio of two finite series in Z.

(Refer Slide Time: 25:02)



It is impossible, impossible to write e raise to the power Z inverse as a ratio of two finite series in Z, Z or inverse does not matter. And therefore, X Z is irrational and concomitant with that is also that if this happened to be the system function of some LSI system, which it can be nothing stops it from being the system function of an SI system.

If it were that system is not realizable, that means, there is no finite hardware or at least no known at least today, there is no known finite hardware or software structure which can realize this. As of date, irrational systems are unrealizable in hardware or software. So, irrational systems are not see we must not confuse the rational with unstable or non-causal or something of the kind, if rationality or irrationality for an LSI system is a property over and above the other properties, causality, stability.

Of course, rationality has a meaning only for LSI system; one cannot talk about rationality or irrationality when the system is not LSI. In fact, not only LSI it must be LSI and its impulse response must have a Z transform. Otherwise, there is no sense in talking about rationality or irrationality. So, now, we have a classification of LSI systems, you have the whole class of LSI systems; they are characterized by the impulse response.

Among them you have a smaller class whose impulse response has a Z transform, that means that the Z transform converges somewhere in the Z plane. And among them you have Z transforms or

LSI systems which have a Z transform, and the Z transform is rational. That is the innermost core, which we are calling rational systems. And those are the systems that we are going to design, implement and build.