Principles of Signals and Systems

Prof. Aditya K. Jagannatham

Department of Electrical Engineering

Indian Institute of Technology, Kanpur

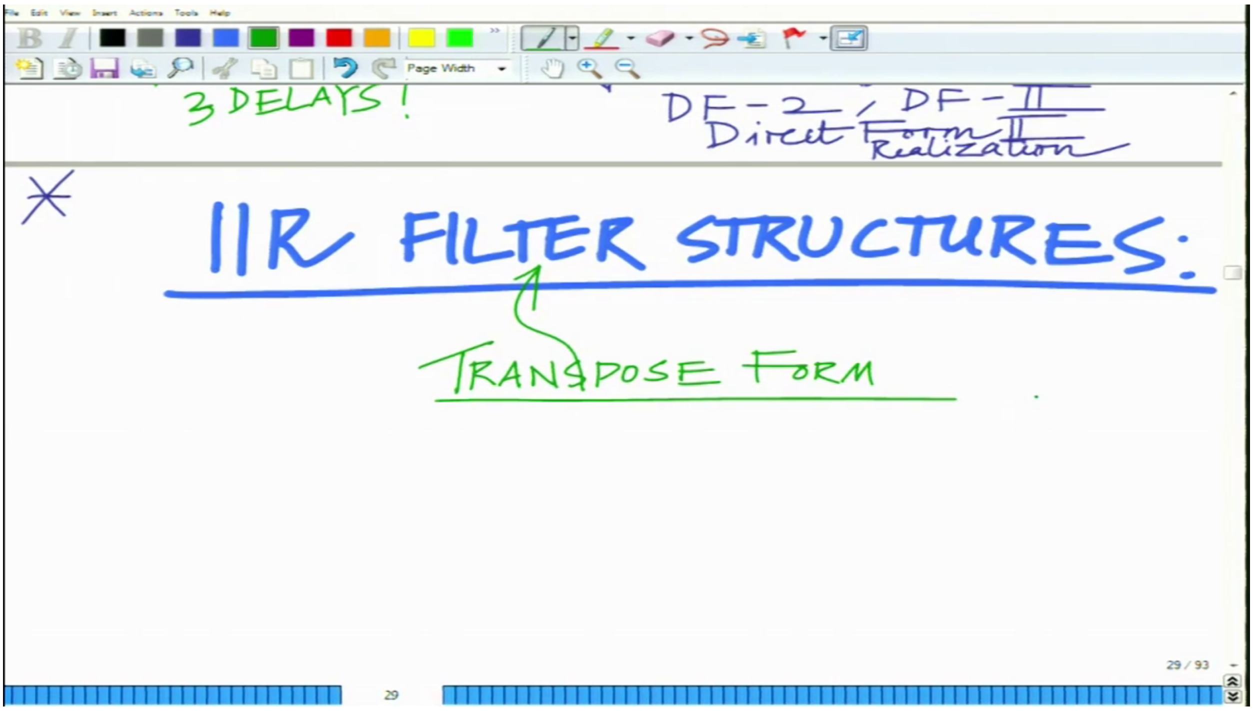
Lecture - 72

IIR Filter Structures: Transpose Form

[noise]

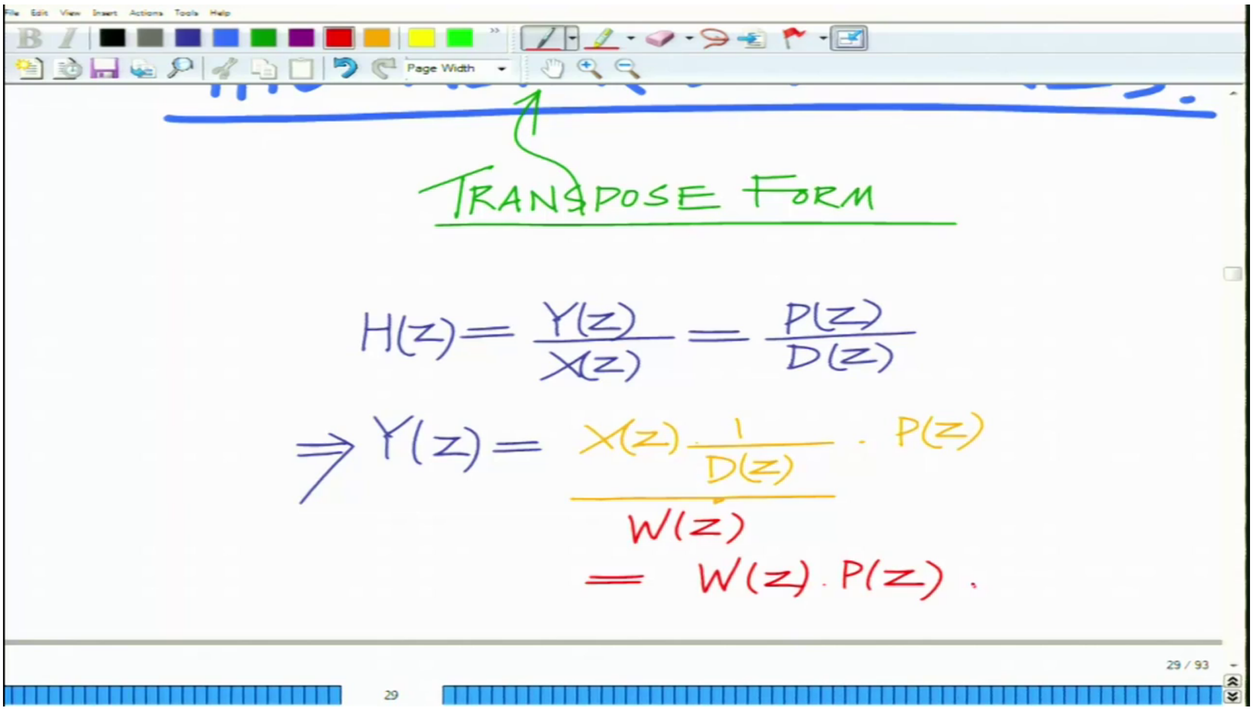
Hello, welcome to another module in this massive open online course, all right. So, we are looking at IIR filter structures [noise] and in this one we will start looking at another filter structure that is the transpose form, [noise] ok.

(Refer Slide Time: 00:26)



So, we are looking at [noise] IIR, [noise] [vocalized-noise] we are looking at IIR filter structures and as part of that [vocalized-noise] we are looking at the [noise] transpose structure [noise] or let us call this the transpose form [noise] [vocalized-noise] and the transpose form can be described as follows. [noise]

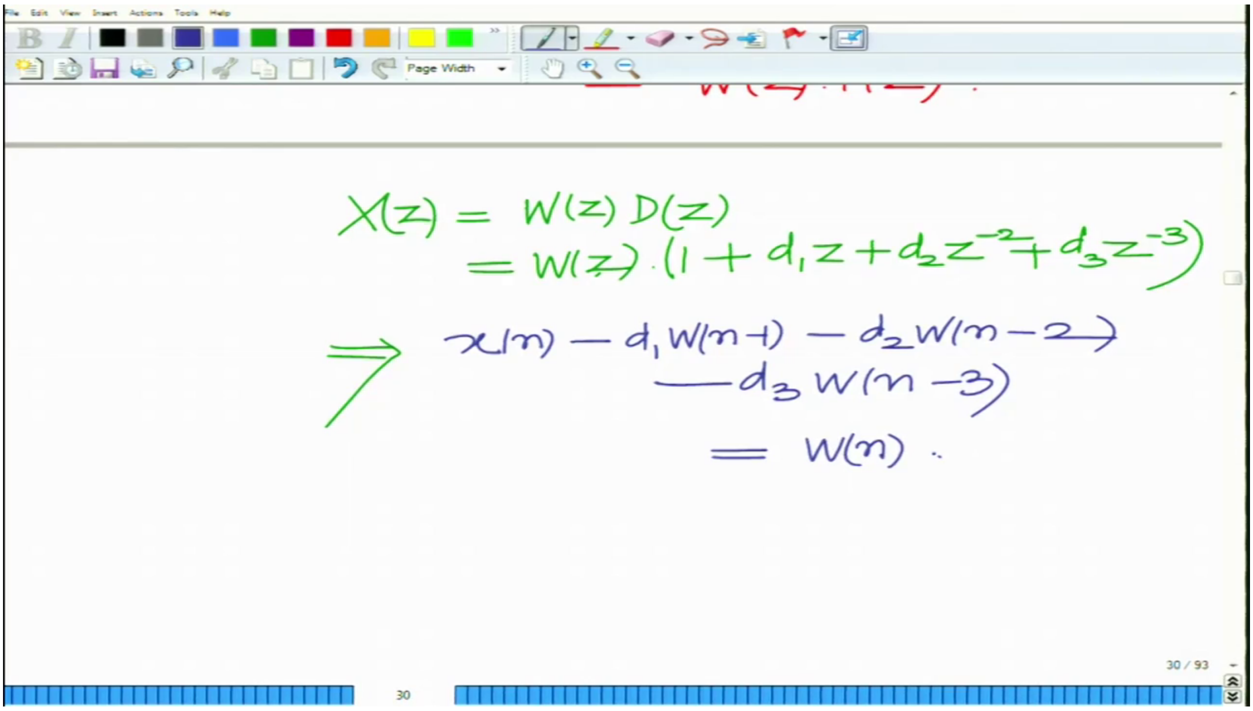
(Refer Slide Time: 01:05)



Now, [vocalized-noise] again consider our H z the transfer function [vocalized-noise] [noise] equals Y z [noise] divided by X z. Let us say that is given as follows [noise] [vocalized-noise] equals let us say similar to what we have considered before [noise] P z divided by D z. So, this implies Y z equals [vocalized-noise] ah X z and now, previously we had written it as cascade of first P z followed by 1 over dz. Now, let us write it as cascade of 1 over D z [noise] followed by P z, [noise] ok.

So, and we will call this X z over D z we will term this as [noise] W z and W z is later on and therefore, I can also write this as W z [noise] times [noise] P z ok. [noise] And therefore, if you look at this we have X z [noise] equals W z [noise] times D z equals W z [noise] times 1 plus [noise] d 1 z plus d 2 z minus 2 plus b 3 z minus 3. [noise] [vocalized-noise]

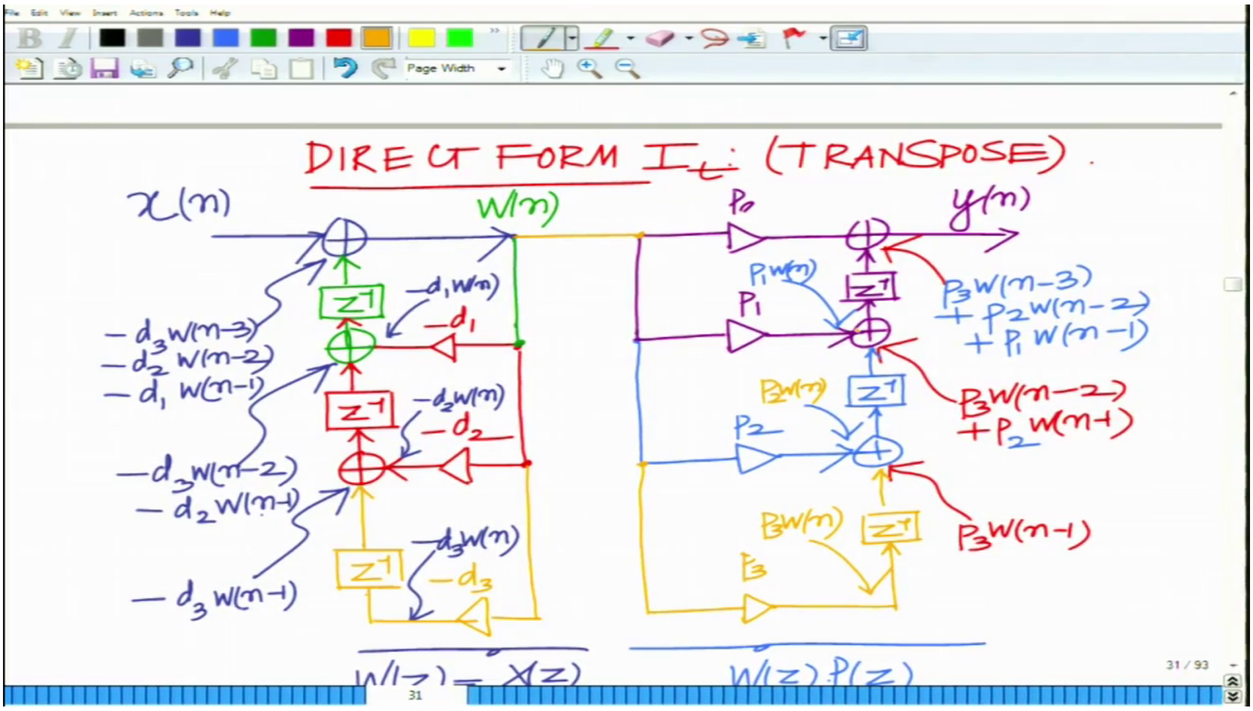
(Refer Slide Time: 02:12)



Which implies I can now, write [noise] converting it into the time domain this implies that [vocalized-noise] all right. I have x of n [noise] minus d 1 W n minus 1 minus d 2 W n minus 2 minus d 3 W n minus 3 [vocalized-noise] this quantity [noise] equals your [noise] W n, ok. [vocalized-noise]

Therefore, I can write this all right. So, we have developed this relation that is x n minus d 1 W n minus 1 minus d 2 W n minus 2 minus d 3 W n minus 3 equals W n ok. And this is the first stage of the cascade ok. [vocalized-noise]

(Refer Slide Time: 03:27)

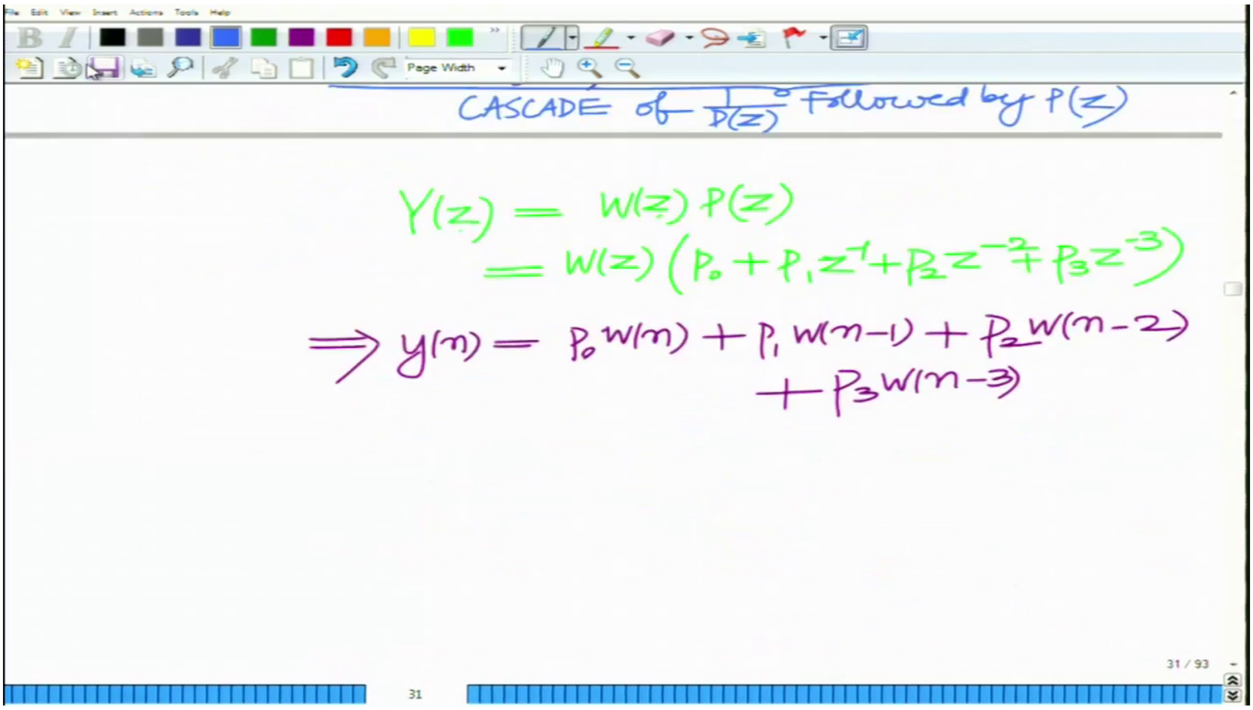


So, I can write this as [noise] I can represent this as x n [noise] an adder [noise] that gives rise to W n [noise] and W n is [noise] in turn being fed as follows. So, I will have [noise] z inverse that is a delay, [noise] this is W n [noise] [vocalized-noise] ok. We have here we will have another adder [noise] and ah then [noise] of course, this has to be multiplied by [noise] minus d 1 ok and ah we will have here, [noise] [vocalized-noise] we will have another delay element [noise] the z inverse ok. [noise] And ah we will have [vocalized-noise] another adder [noise] another gain [noise] [vocalized-noise] minus d 2 ok [noise] and ah [noise] finally, we will have another [noise] element here this is an adder ah this is a delay and ah here [noise] of course, there will be another gain [noise] that will be [noise] ah minus d 3 [vocalized-noise] and ah this will be again ah [noise] all right this is [vocalized-noise] minus d 3. [noise] And ah therefore now, if you see and this W n we will complete this later is being passed forward.

So, now, if you see [vocalized-noise] this here this element at this point we have minus d 1 W n [noise] and ah at this point we have minus d 2 W n and at this point we have minus d 3 [noise] W n and you can see here I have minus d 3 W n [noise] minus 1, [vocalized-noise] ah at this point I have minus [noise] d 3 W n minus 1. Further delayed, so that will be minus 3 d 3 W n minus 2 minus d 2 [noise] W n minus 1 [vocalized-noise] and at this point finally, I will have minus d 3 [noise] W n minus 3 minus d 2 W n minus 2 minus d 1 [noise] W n minus 1 added 2 x n that produces your wire, ok. So, that is basically [vocalized-noise] the first part. So, this part is implementing your [noise] W z equals [noise] X z over [noise] D z [vocalized-noise] ok.

So, now, we have this quantity W n all right and now, from W n we have to generate wire all right. So, and we know that W z into P z is Y z, ok. So, using that the using this system we generate ah that is that leads to the output Y z ok.

(Refer Slide Time: 07:02)



So, we have Y z [noise] [vocalized-noise] equals [vocalized-noise] W z times [noise] P z which is W z times ah that will be P naught [noise] plus P 1 z inverse plus P 2 z minus 2 plus P 3 z minus 3. [noise] And ah that will be and this implies basically that y n equals simply P naught [noise] ah W n plus P 1 W n [noise] minus 1 plus P 2 W n minus 2 plus [noise] P 3 w [vocalized-noise] n minus 3 and that is the forward path.

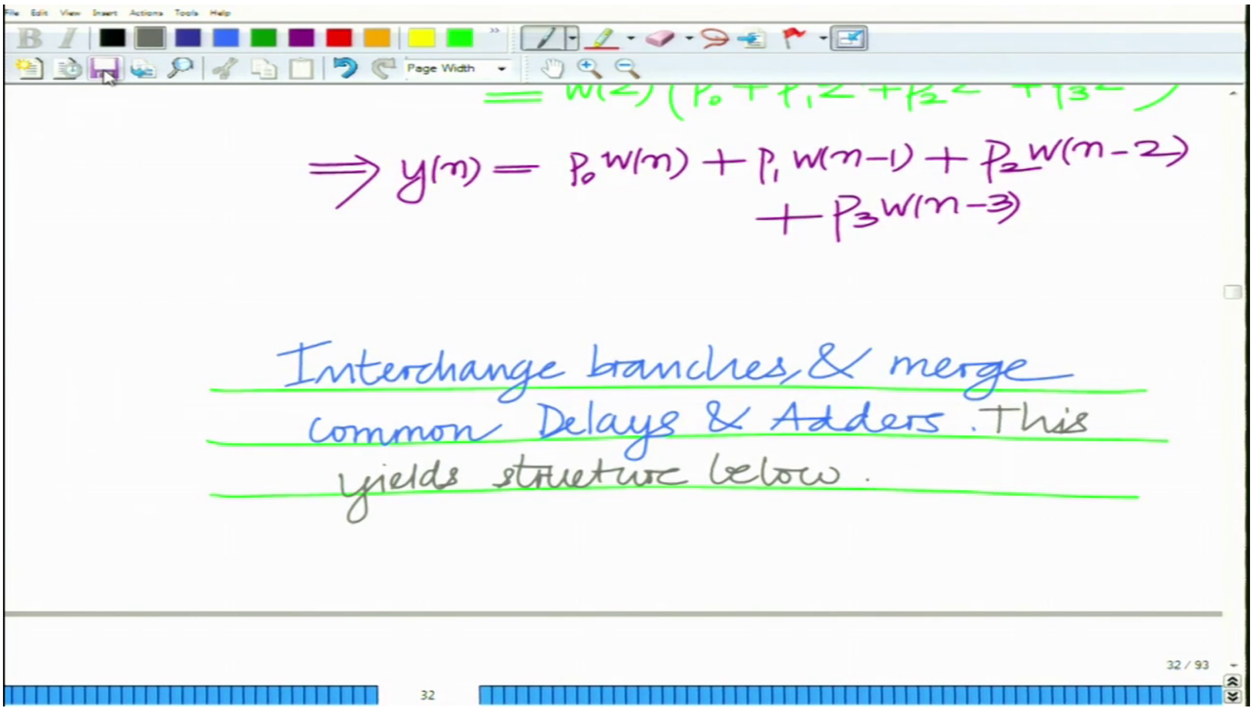
And now, you can see [vocalized-noise] W n [noise] send it through gain P naught [noise] [vocalized-noise] ok, [noise] it is a adder that gives me [noise] w y n. [noise] Now, this [noise] I have to [noise] also tria central gain P 1 and this will be delay z 1 z inverse all right and that will be give that again you have another adder here ok and that will be [vocalized-noise] that will give you ah [vocalized-noise] this is P 1 W n. And now, I will [noise] send that through another gain that is [noise] P 2 [noise] [vocalized-noise] and this is ah well another delay over here, [noise] [vocalized-noise] and finally, you send it through another gain stage that is [noise] [vocalized-noise] and another delay. [noise] So, that will give me the gain is P 3. [vocalized-noise]

And ah so what you have here [noise] is basically you can see [noise] let me just mark this ah at this stage you have P 1 [noise] W n at this stage you have P 2 [noise] W n and at this stage you have P 3 [noise] [vocalized-noise] W n. Here at this point [noise] you have P 3 [noise] W n minus 1 at this point you have [noise] P 3 because [vocalized-noise] its passing through two delays P 3 W n minus 2 plus P 1 W n minus 1 and at this stage you have [vocalized-noise] ah P 3 W n minus 3 [noise] plus P. I am sorry this is P 2 [noise] W n minus 1 [vocalized-noise] all right, so plus P 2 W n [noise] minus 2 plus [noise] P 1 W n minus 1, [noise] ok.

And ah so [noise] what you have over here [noise] ah will be and now, you are adding this to P naught W n, P naught W n, so that gives you Y n. [noise] So, that basically completes and this part basically implements the [noise] W z into P z. [noise] So, this is basically your cascade or first 1 over D z with W z. So, you can just write this as [noise] cascade [noise] of 1 over D z [noise] followed by [noise] P z, ok. [vocalized-noise] [noise]

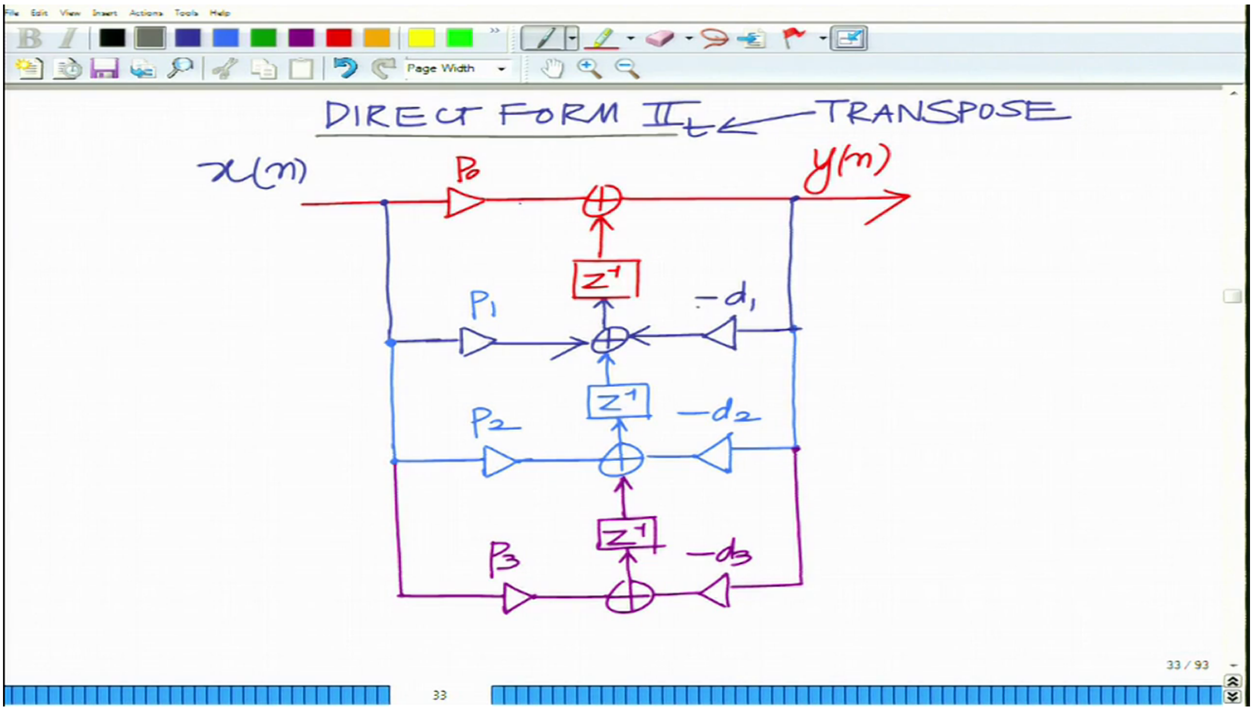
And now, [noise] once against remember similar to the ah similar to what we did for the direct form what we want to do here is we want to now, ah interchange [vocalized-noise] these two branches. And once we interchange these two branches we can see that the resulting delays and in this case in fact, even the adders can be combined for both the branches. So, once you combine them now then you will get the representation with the minimum number of delays and adders. So, what we are going to do again here [noise] is we will interchange [noise] branches and merge the common delays and adders ok.

(Refer Slide Time: 11:59)



So, [noise] [vocalized-noise] ah what we will do is we will [noise] interchange [noise] [vocalized-noise] and we will merge, [noise] we will merge the [noise] [vocalized-noise] common, merge the [noise] common [noise] we will interchange. And and this yields [noise] this yields [noise] structure below [noise] this yields the structure that is shown below [vocalized-noise] which is basically ah [noise] [vocalized-noise] and you can derive this I am not going to explicitly show the steps, [vocalized-noise] but you can do this as an exercise. [noise]

(Refer Slide Time: 13:11)



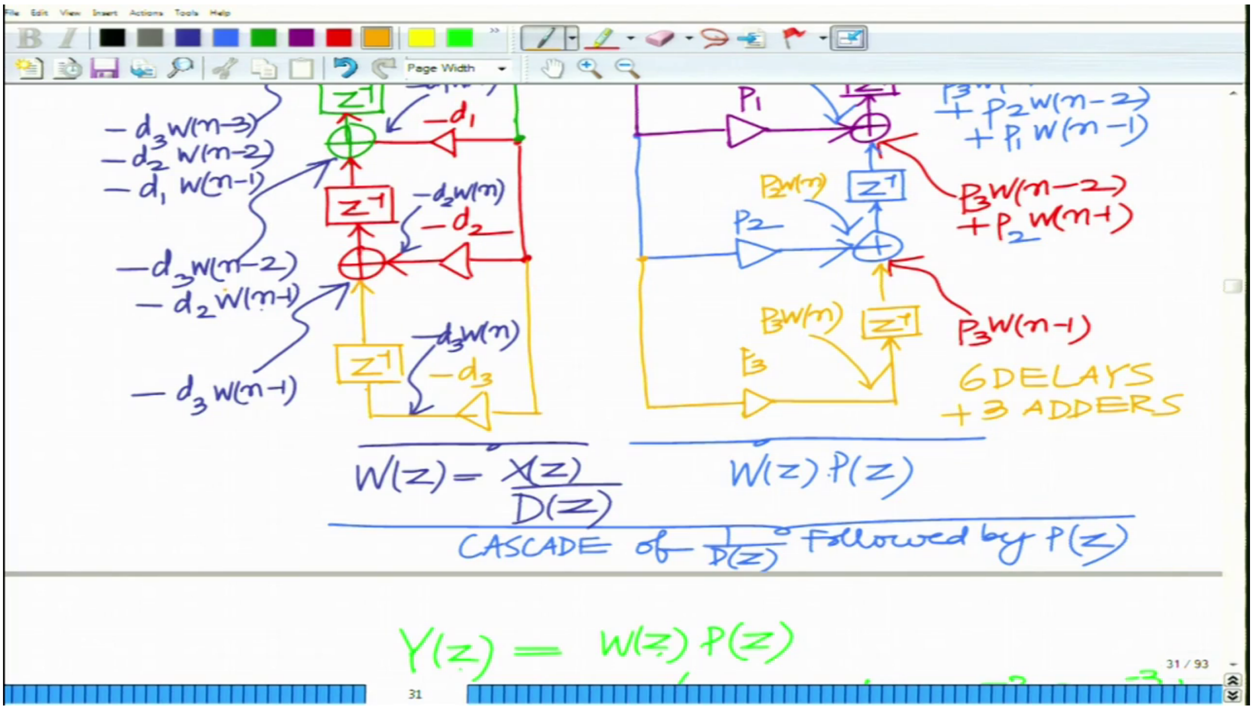
So, this will give you [noise] P naught, [noise] this is the element [noise] gain, this will give you y n ok [noise] and here you will have a delay, [noise] z inverse [noise] all right and ah [noise] another adder. [noise] And from this you will have [vocalized-noise] [noise] minus d 1 [noise] and here you will have [noise] minus d 2 [noise] [vocalized-noise] and this one is minus d 3 [noise] that also be one delay, [noise] [vocalized-noise] ok. And ah [noise] I am sorry this is P 1 [noise] and this will be P 2 [noise] ok.

Remember you are interchanging the branches. So, this will be minus d 2 [noise] [vocalized-noise] ok and finally, you will have [noise] one more [noise] [vocalized-noise] and this will be gain will be P 3 [noise] and this gain is minus d 3. [noise] [vocalized-noise] And what you have over here ah is basically this is termed. So, this is the alternative representation this is termed as the ah direct form. So, previously what we so, this is formed as this is termed as the direct form II for the transpose. [noise]

So, what we have over here is the direct form one, this is termed as the previous one is termed as direct form I transpose. [noise] We have seen previously direct form I and direct from 2. [noise] So, this is the direct form transpose [noise] ok. So, this is the transpose [noise] structure [vocalized-noise] [noise] that is denoted by this t in the subscript. And what we have over here is this is your direct form II transpose [noise] [vocalized-noise] and this t in the subscript [noise] this denotes the, [noise] this denotes the [noise] transpose, ok. [vocalized-noise]

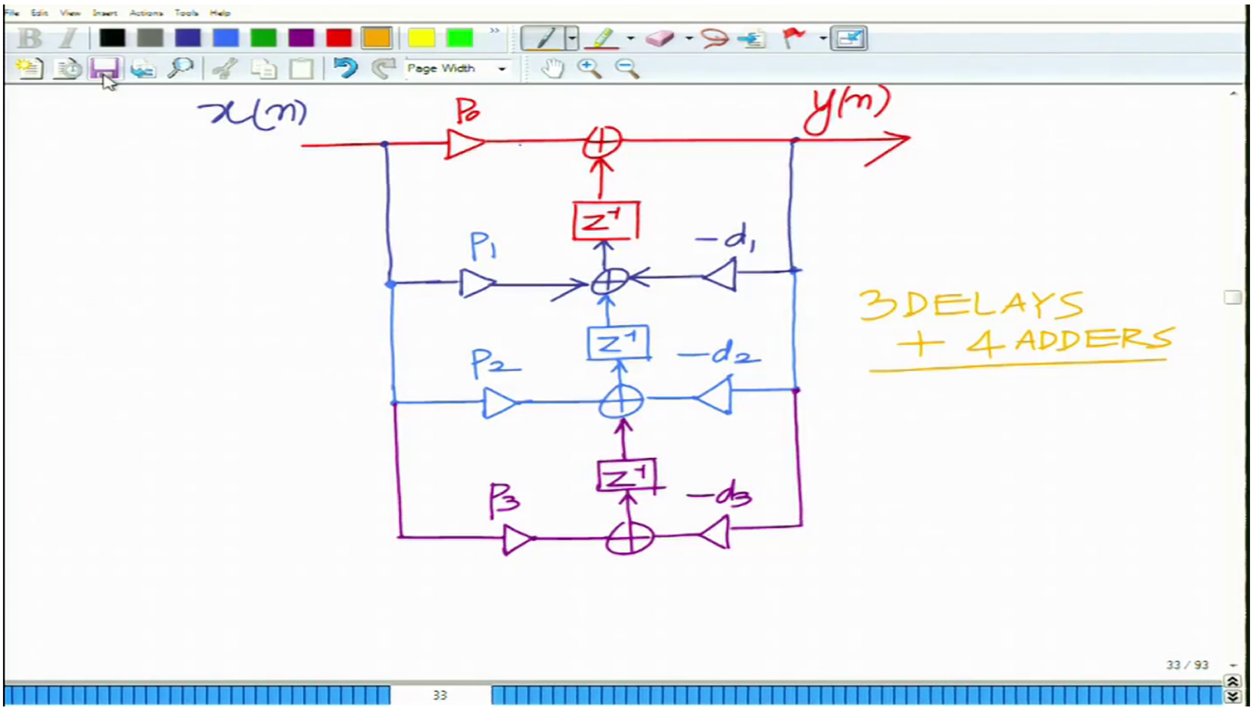
And ah [vocalized-noise] what we have is basically you can now, see that we have the minimum number of delays [noise] because previously in the direct for because previously in the direct form I transpose we had ah you can see we basically have 6 delays and we have 7 adders. If I am not if you can count correctly, so this has [noise] not 6, this is this is not an adder you do not need this actually. [noise] So, [noise] you just join this [noise] here and what you have [noise] is basically you can see you have 6 delays and 3 adders. So, this has 6 delays [noise] plus 3 adders, 3 adder [noise] elements. [vocalized-noise]

(Refer Slide Time: 17:20)



But if you look at the, [vocalized-noise] but if you look at the transpose if you look at the direct form II transpose that has only 3 delays [noise] plus it has only 4 adders. [noise] [vocalized-noise]

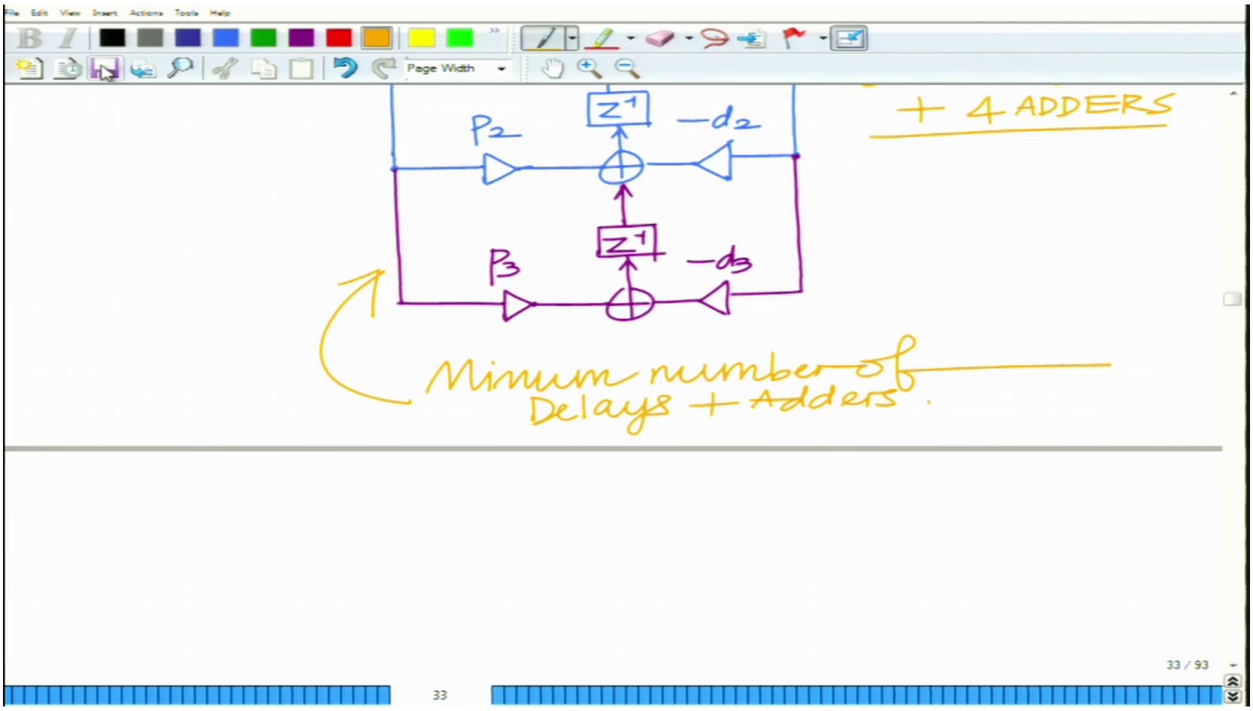
(Refer Slide Time: 17:43)



So, this has it is, so the direct form II transpose [vocalized-noise] has fewer number of delays similar to what you saw in the direct form, ah direct form II correct. This is a transpose and this has the few this is fewer add [vocalized-noise] all right, all right. So, in comparison to the direct form I transpose this is as fewer delays that is from 6 the number of delays reduced to 3. In addition the number of adders [vocalized-noise] is also significantly decreased from ah 6 to 4 adders.

So, this is the minimal representation in terms of the number of delays plus the number of adders, ok. [vocalized-noise] So, this is the [noise] this results in the minimum number of delays, [noise] minimum number, [noise] [vocalized-noise] this has the minimum number of delays and adders [noise] ok. So, that is the [noise] [vocalized-noise] direct form. [noise] So, that describes basically the direct form II transpose, [vocalized-noise] all right. [vocalized-noise]

(Refer Slide Time: 18:32)



So, in this module what we have seen is we have looked at the direct form II, ah the direct from I its transpose structure and also the direct form II its transpose structure and we have seen that the direct form II transpose results in the minimum number of adders and delays. So, let us stop this module here and continue in the subsequent.

Thank you very much. [noise]