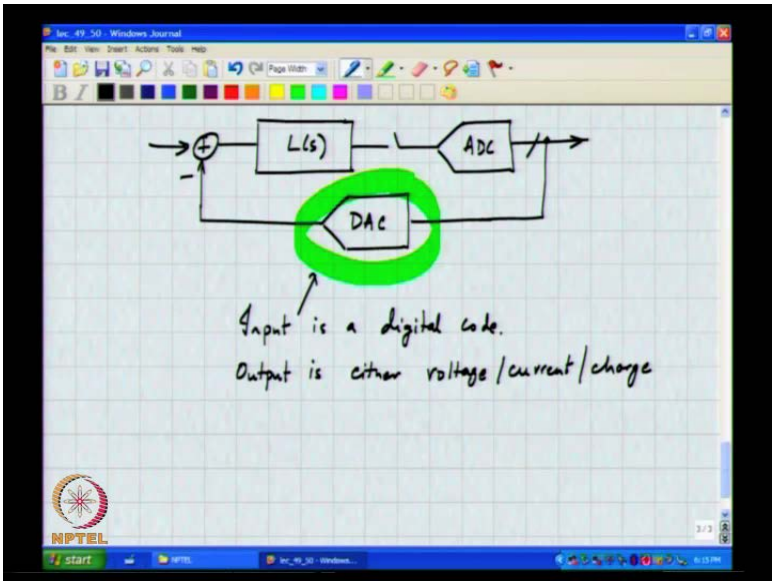
VLSI Data Conversion Circuits
Prof. Shanthi Pavan
Department of Electrical Engineering
Indian Institute of Technology, Madras

Lecture - 50
DAC Basics

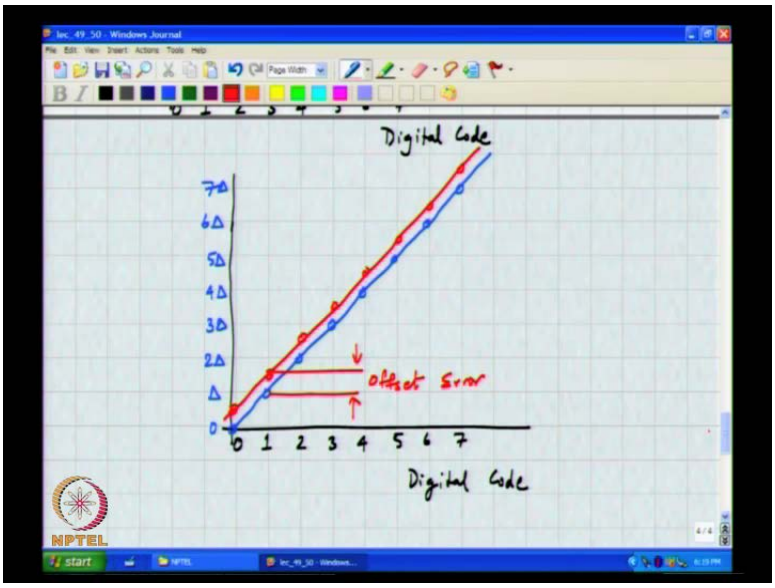
This is VLSI Data Conversion Circuits, lecture 50, today we will begin the study of another important piece of the delta sigma loop, and use this as an excuse to study D to A converters, so we will be looking at this.

(Refer Slide Time: 00:30)



Now, as the name suggests the input is a digital code, the output is either voltage or current or charge.

(Refer Slide Time: 01:53)



And if one plots the digital code on the x axis, and the analog output on the y axis will get ideally you are suppose to get a characteristic like this, please note that these inputs actually do not make any sense, the code can only be an integer all. And delta is called the step size just like in a A to D converter.

This is the positions of the x and y axes are interchange, and this represents a 3 bit D to A converter delta is the LSB or step size of the duct, now just like how the thresholds of an A to D converter can be in error due to various factors. For instance, due to mismatch in analogous manner, the steps of a D to A converter can also be in error, so how do you think erroneous characteristic will look like, will it be changes in the x axis or in the y axis.

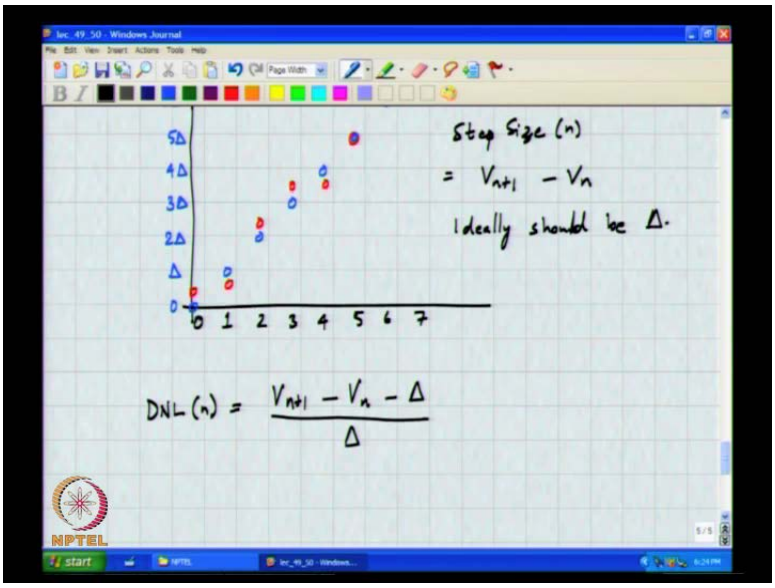
Student: Y axis.

Only in the

Student: Y axis.

Y axis, so the common kinds of errors is a very similar to what we discussed with A to D case, if one draws, joins the output levels of an ideal D to A converter, you must get a straight line. In practice one common kind of error is when you join these points, you get a straight line; however, this straight line is shifted up or down from the ideal line, and this is called the offset.

(Refer Slide Time: 05:56)



Another kind of error, again all this should be very familiar given to expose to A to D converters is when you join these points, you get a straight line; however, the slope is not you know delta for every change in the input code, it is more than the delta. That has been coated, for example on the data sheet, so this is like having a larger delta that is all, so this is called gain error does it make sense.

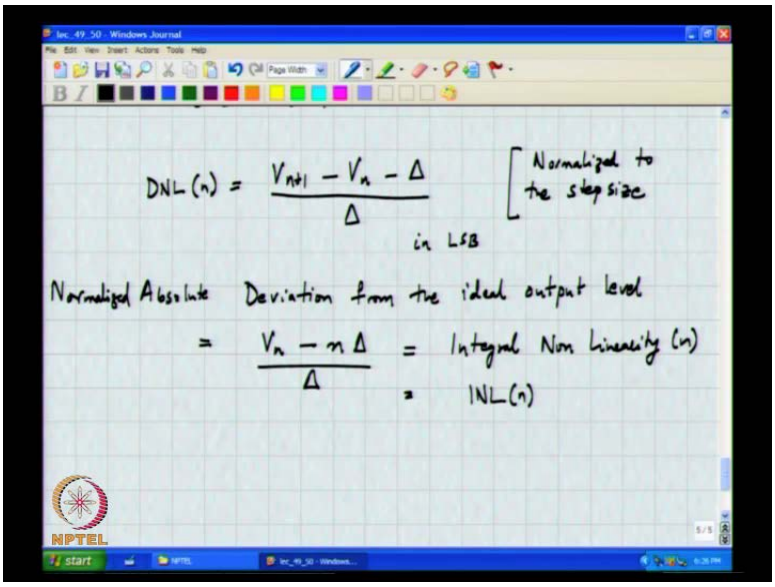
So, gain and offset errors are not really that problematic in a system, because both of these would be calibrated out anywhere, what is important is that the straight line remains a straight line. Of course, that is not true either, so the outputs may be deviating from their ideal values in this fashion, and therefore as you can see the step size is not the same from code to code.

And therefore, the step size at code n, if we call the output for code as V n, and the output for code n minus 1 is v n minus 1, then this is the or rather the step size of code n is nothing but, v n plus 1 minus v n. And ideally should be delta in practice it is not delta it is either larger or smaller, so how can you quantify the deviation from the ideal step size.

We know that they are all should be about delta, and we are actually interested in the deviation from delta, so analogous to the A to D converter case, the deviation from the ideal step size, so that is v n plus 1 minus v n minus delta, which has absolute units of in this particular case volts. If you want to talk about compare different converters, it make

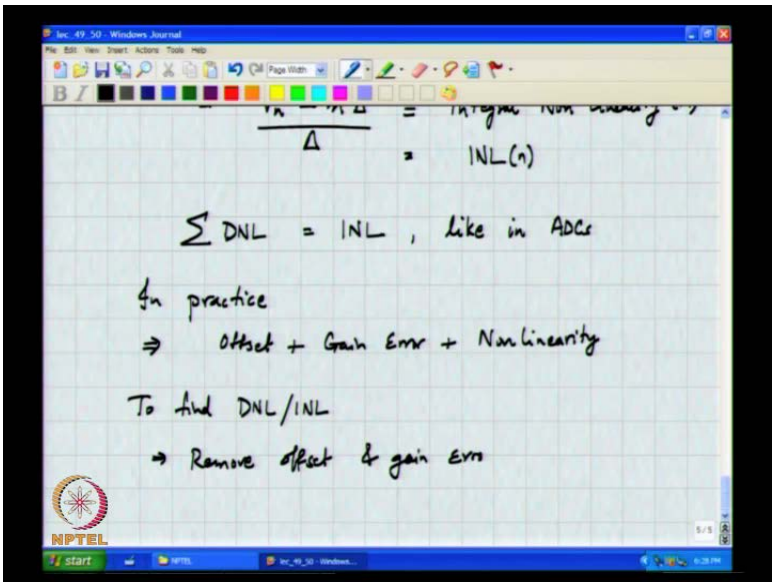
sense to not talk about the absolute deviation, but the relative deviation in which case you normalize it to the step size, and this is called the DNL at code n.

(Refer Slide Time: 10:56)



And similarly, so this is normalized to the step size the units are DNL of n is in LSB, on the other hand the absolute deviation from the ideal output is also of interest, and that is simply is v n minus n delta. And again it does not make sense to talk about absolute deviation or the dimension quantity, so you normalize this, and this is called the integral nonlinearity or INL of n.

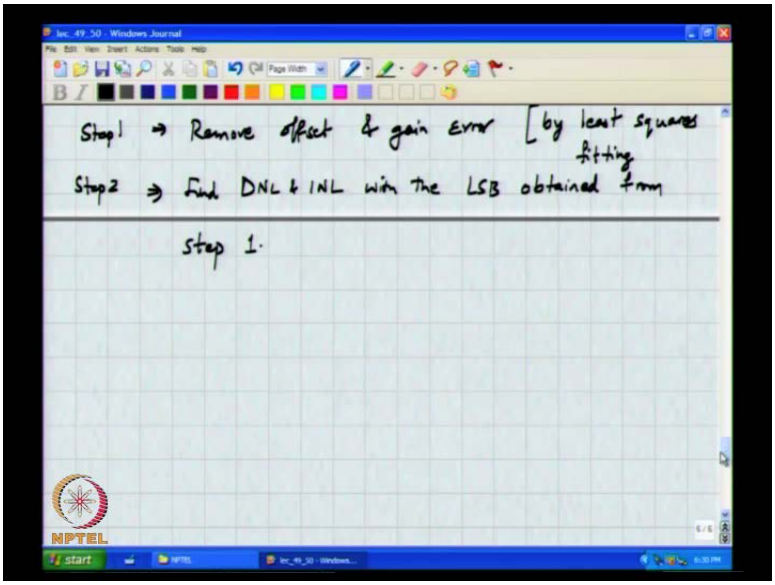
(Refer Slide Time: 13:04)



And the running sum of the DNL is the INL just like in ADC's, now these in practice is not, as if the converter will have a just offset or just gain error or only DNL or INL, the whole thing will become as a cocktail. So, in practice a converter will have offset plus gain error plus nonlinearity which is characterized by DNL and INL, so if you want to find the DNL and INL of a converter, what should you do first.

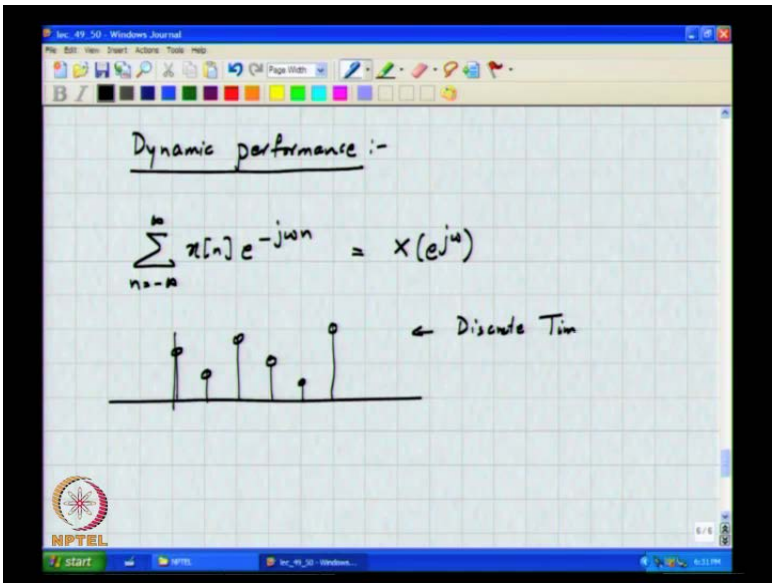
Student: minimum speed.

(Refer Slide Time: 14:52)



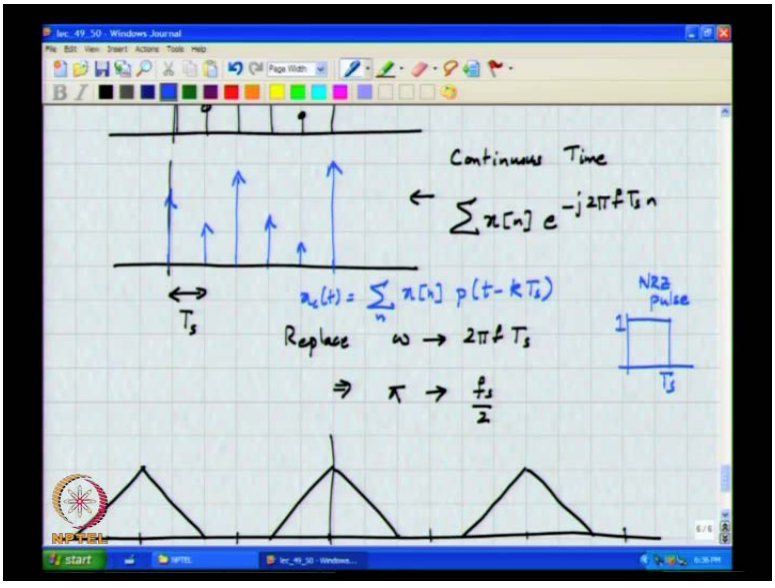
So, the way of finding DNL and INL is first remove offset and gain error and then, so by best fit, for example and then from this you will find the actual the I mean the true l s b size, which will result in a closest fit to the D to A converter characteristic. And once you remove offset and gain error, you can find the DNL and INL with the step size obtained from step 1, the LSB obtained from does make sense, the next thing is dynamic performance the input to the D to A converter is a sequence.

(Refer Slide Time: 16:20)



And it is discrete time Fourier transform is nothing but, e to the x of n e to the minus j omega n sum over, so this is nothing but, x of e to the j omega. Now, conceptually what the D to A converter is doing is taking the sequence of discrete time impulses, and converting it into a continuous time signal.

(Refer Slide Time: 17:30)



And therefore, one can think of the whole operation as replacing the discrete time impulse with a dirac delta pulse, and the spectrum, therefore of the continuous time signal is nothing but, x of n this is the sampling rate of the D to A converter e to the

minus j 2 pi

Student: A to D

F times f T s times n, this is the Fourier transform of the continuous time signal does it make sense . So, how do you relate the discrete time Fourier transform to the continuous time Fourier transform.

Student: Minimize ((Refer Time: 19:08))

All that you need to do is replace

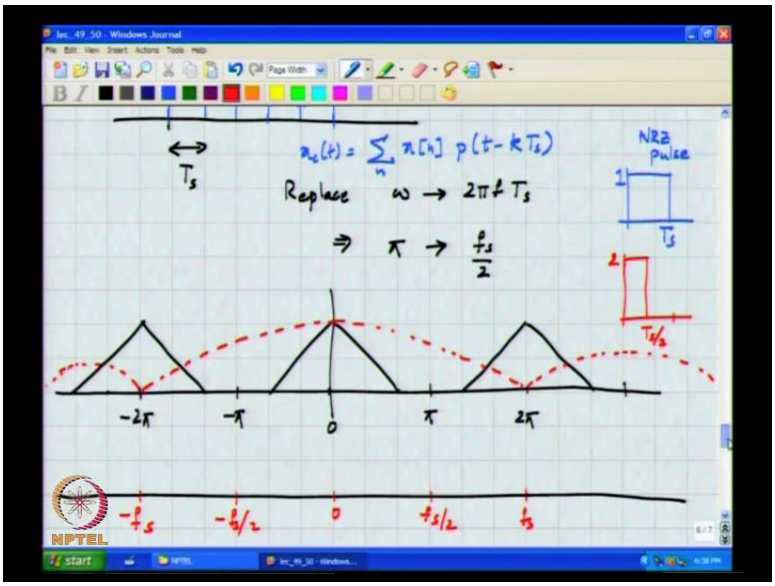
Student: Minimum Fourier transform

Omega by 2 pi f times

Student: T s

T s, so as usual omega equal to pi corresponds to f s by 2

(Refer Slide Time: 19:53)



And therefore, if the discrete time spectrum look like this, and I am sorry discrete time spectrum look like, then all that one needs to do is change the axis to 0 f s by 2 f s unfortunately, generating dirac delta pulses is not realistic in practice. So, every DAC has got a pulse shape and we have discuss this before in the context of a delta sigma

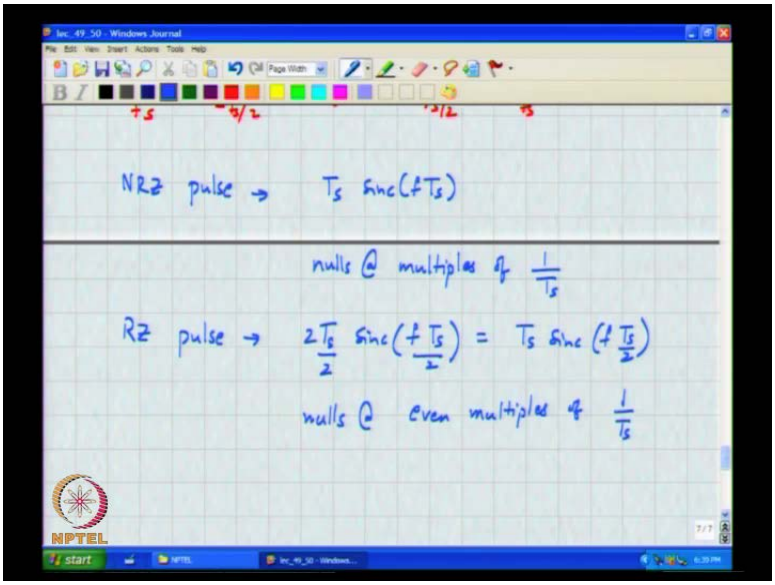
modulator, common pulse shapes are the NRZ pulse, where the output can be thought of as x of n.

And the time domain can be thought of as x of n times p of t minus k times T s, the output in continuous time is this, where p of t is 1 from 0 to T s, so can you comment on the spectrum now. So, you can think of this as a taking the continuous time waveform, which is at sequence of dirac impulses and passing them through a low pass filter, whose impulse response is rectangular.

If the impulse response is a rectangular it follows that, the Fourier transform is sinc function, and it has null set multiples of 1 by T s which is f s. So, in other words the spectrum of the NRZ DAC will be something like this, on the other hand if you had the, so called return to 0 pulse, what happens to the spectrum now.

Student: Once for ((Refer Time: 24:11))

(Refer Slide Time: 24:23)



So, for an NRZ pulse the filter is of the form T s sinc f times T s. So, this has null set multiples of 1 by T s, for a return to 0 pulse, it is 2 T s by 2 sinc f times T s by 2, which is the same as T s sinc of f T s by 2 which means this has null set.

Student: ((Refer Time: 25:22)) 1 by 2 by

F plus at even multiples of 1 by T s, and that makes physical sense. So, the spectrum to

the NRZ, so this is the NRZ filter if you saw, if you like the RZ filter will only have will look like that, so these are often called the images of the signal. And as you can see the images are attenuated to a much larger degree by a, when you have a NRZ DAC pulse shape rather than a return to 0 DAC pulse shape.

And that makes sense because a return to 0 pulse shape has got much larger transitions, and these transitions occur twice every period you understand. That means, that the jumps going up and down are much larger in the return to 0 case which also means that, then it must have much higher strengths of high frequency components, so the spectrum actually makes sense. So, now, the question is why would anybody choose a return to 0 DAC pulse, given that there is the signal has got lot more high frequency content, do you understand.

When, you use an NRZ DAC pulse shape the signal has got the images are alternated to a larger degree, the height of the pulse is smaller and there are no transitions in the middle. So, one obvious question that could be asked is if everything is, so nice about the NRZ pulse, why even talk about a RZ DAC, yet we have gone and seen the applications of this in a continuous time delta sigma loop, we even discussed jitter and such. So, the question is yeah, so why do you think a return to 0 pulse makes any sense at all.

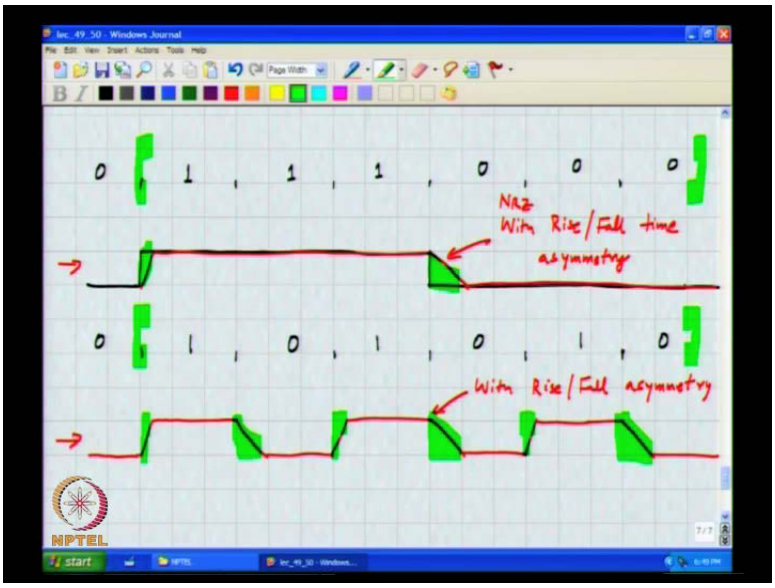
Student: ((Refer Time: 28:32)) due to excessive.

In the context of a delta sigma loop as he points out, definitely you know return to 0 pulse is got is a lot, when use a return to 0 DAC you are a lot less sensitive to.

Student: Axis loop delay.

Axis loop delay, but there is another important and not. So, obvious point that I will make now no real NRZ or RZ DAC will have perfectly sharp edges, so they are always be of finite rise time and a finite fall time.

(Refer Slide Time: 29:16)



Now, let us consider a sequence say 1, 1, 1, 0, 0, 0 let us assume that it was 0 throughout, now an NRZ DAC waveform ideally would look like this, and now the rise and fall times A are not 0 first of all, and B could be a symmetry. This is also a very common thing for example, if you use a I mean if you use a if you look at a C MOS inverter for example, the rise time is always if you if you not size the devices properly, you will find that there is always a difference between the rise and.

Student: Fall

Fall times. So, it is not uncommon to have NRZ DAC pulses, where the NRZ pulse has different rise and fall times, I am going to grossly exaggerate the rise and fall time asymmetry now, and therefore the actual waveform looks like this. Now, let me consider another waveform another sequence, which is 0, 1, 0, 1, 0, 1, 0 and with rise time and fall time asymmetry the waveform would look like this. Now, if the system is perfectly linear consider this the input to this, the input sequence to the system is 3 1's followed by 3 0's, and here it is 1 0 1 0 1 0.

So, if the system has perfectly linear, since both these sequences have the same average values is that something that you agree with.

Student: Yes.

The average value of 1 one 1 0 0 0 is the same as the average value of.

Student: 1 0 1 0.

1 0 1 0 1 0. So, if the DAC is perfectly linear, then the average value of the red waveform here should be the same as the average value of the red waveform drawn below, now can you comment on the average values of the 2 waveforms.

Student: ((Refer Time: 33:56)) from bottom of this line.

Student: Bottom of this line.

The average value of the lower 1 is higher, and why is that.

Student: 1 2 2 by 2.

Student: Fall time is fall time is more, if it is a square wave, e r remains the same and a rise times and fall times are equal, the losses the area in this portion will be compensated by this.

Absolutely.

Student: Here the fall time is much higher.

Correct.

Student: So, you have for every rise time and fall time you have some delta extra error adding.

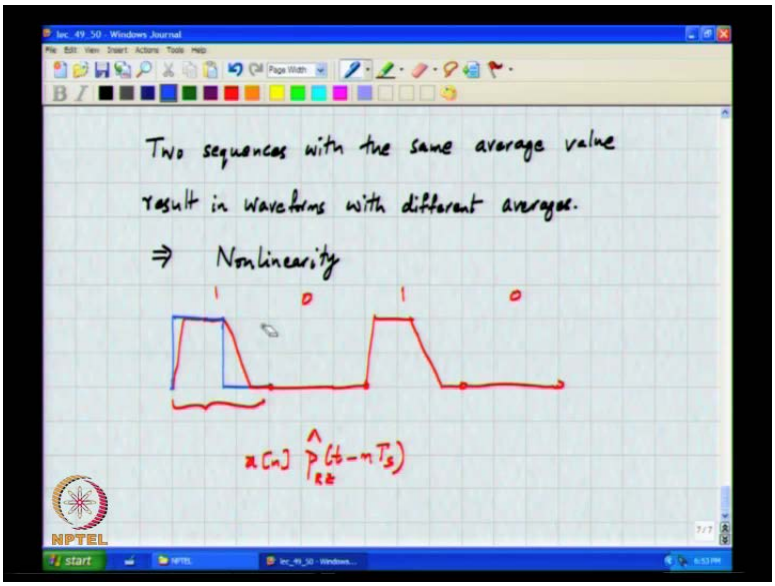
So, it is a very good observation that if the rise and fall times were exactly equal, then when you compute the average the loss in area here would be compensated exactly by the area increased there. However, because the rise and fall times are not the same, these areas are not compensated, so if you have more falling edges and rising edge pairs in a given amount of time, you keep accumulating that error.

In other words the average value of these 2 waveforms are not the same, and I mean this is an example I chose, but you can clearly see that, even the average value of the waveforms depends on the sequence that is being put in. So, 2 sequences having the same average, do not result in 2 waveforms with the same average, so what is this, can you give me one word to characterize this property.

Student: Nonlinear.

It is non-linear.

(Refer Slide Time: 36:05)



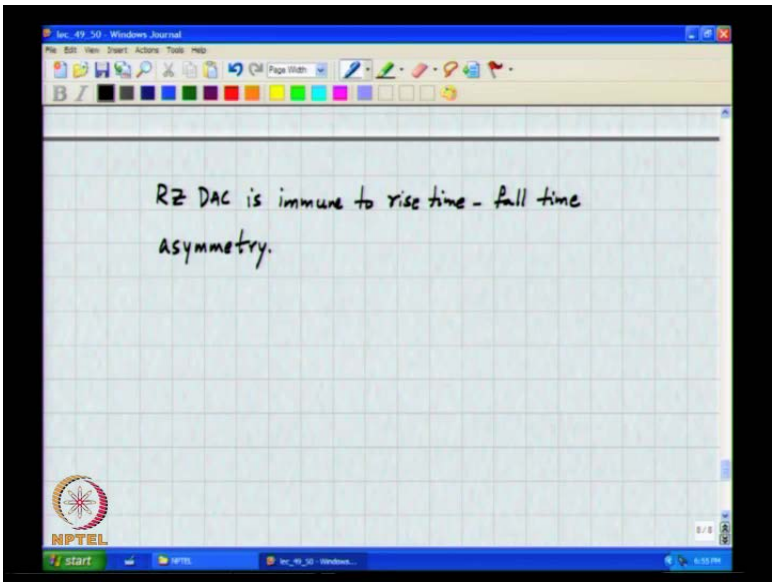
Let me put that down here 2 sequences with the same average value result in waveforms with different averages, and this is nonlinearity, and this is all that is needed to establish that rise and fall time asymmetry causes nonlinearity. Of course, to characterize this very carefully I mean you can do it, but the bottom line is that using an NRZ DAC pulse shape with rise and fall time asymmetry will result in nonlinearity.

On the other hand with a return to 0 pulse, the height of the pulse must be twice the duration should be half, and even if there a significant rise and fall time asymmetry. So, whether it is a 1 0 1 0 pulse or a 1 1 1 0 0 pulse, I mean whether it is a 1 0 1 0 sequence or a 1 1 1 0 0 0 sequence we see that the average remains the same. It is just that the pulse shape, which was ideally a rectangle of height 2 and duration T s by 2 is now altered, all that it means is that it is a linear phenomenon where the filter applied to the ideal impulse strain, is not what you thought.

It was slightly different which means that the high frequency response changes a little bit, but this is not a non-linear phenomenon, does it make sense. So, this waveform can always can still be represented as x of n to p of t minus n T s instead of having p NRZ p RZ, it is some p hat r z that is all. Ideally you would have got you would have expected

this, now instead of having the blue pulse you have the red pulse, but it is still a linear system.

(Refer Slide Time: 40:05)

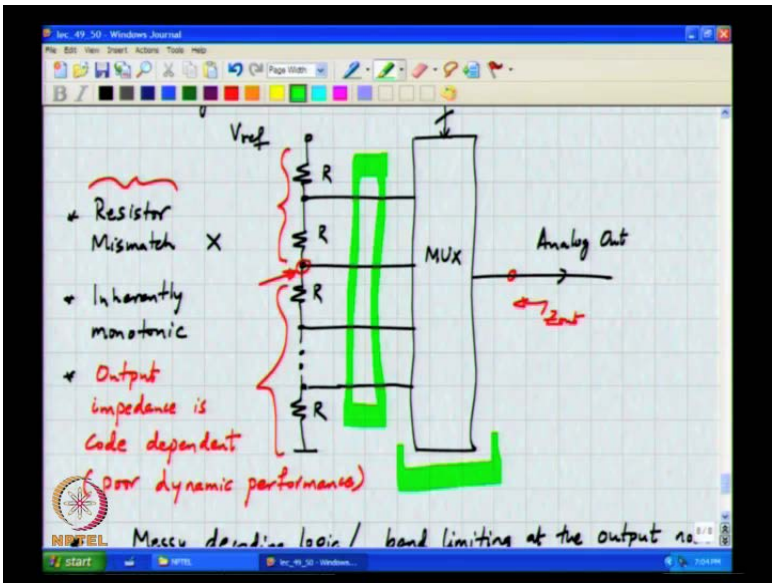


So, an NRZ an RZ DAC is immune to rise time fall time asymmetry, unfortunately a return to 0 DAC has other issues namely, in the context of a delta sigma modulator. It is got problems with respect to jitter, as well as the loop filter now has to handle signals which are I mean who peak amplitudes are twice when compare to the NRZ case, but we see that the return to 0 DAC has an advantage over a NRZ DAC.

This becomes a particularly important when, you are running this at very high speeds where the rise time becomes a non negligible fraction of a entire clock period or in cases where the precision required is, so high. That even a small change in the rise and fall time or rather small mismatch in the rise and fall time can give rise to components which are significant when considering the very low restoration that one needs.

So, in oversample converters we also saw example of a third pulse shape, which is again fairly commonly used and that is the exponentially decaying pulse shape. Now, one way of implementing these DAC's, let us just first consider a simple NRZ DAC, the output of the flash is a digital code.

(Refer Slide Time: 42:41)



And one can think of many ways of converting this digital code into an analog output, so what is the simplest way you can think of converting, so easiest thing you can think of perhaps is a resistive sting. You have a reference, you have a bunch of nominally identical resisters, and you have some kind of mux is an analog mux now, which has many inputs, and the digital the digital code selects, which one of these perhaps get to the output. So, what do you think are the problems with this approach.

Student: Resistance will not be an ((Refer Time: 44:55)).

Of course, like resistor mismatch will mean that the steps are not exactly equal, can you comment on what do you call the monotonicity of this DAC.

Student: Monotonic.

It is monotonic, this is an inherently monotonic, can you comment on the output impedance of this DAC.

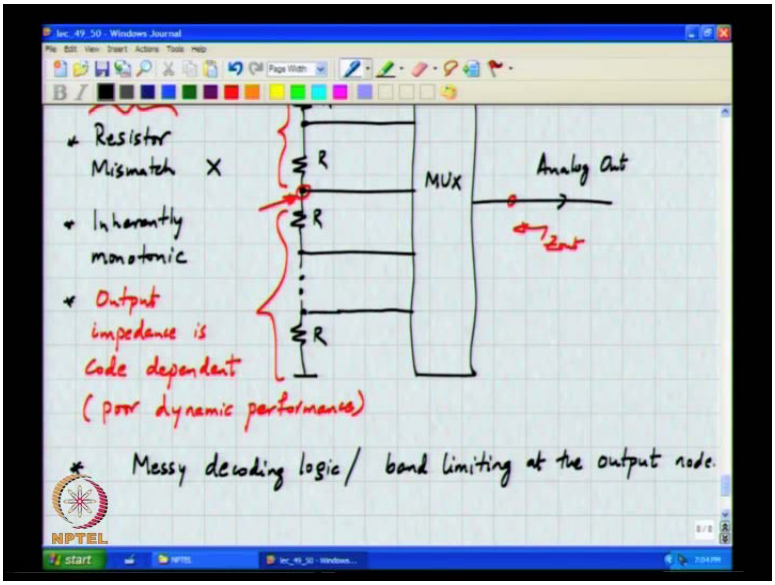
Student: Difference on the difference of the ((Refer Time: 46:15)).

As, you can see depending on the digital code, the output impedance is this resistance in parallel with the resistance below, so the output impedance is code dependent. So, this can be I mean a big problem, from a point of view of nonlinearity, I mean the output waveform, therefore will have different time constants depending on the input code. So,

if you are interested in the entire waveform or the spectrum of the waveform, you should be prepared to accept a lot of distortion, simply because the time constants keep changing depending on the code.

However, if you are only interested in the settled value, then you do not need to bother, you are only have to worry about the static aspects of the entire DAC, which is the resistor mismatch.

(Refer Slide Time: 47:43)



So, the fact that the resistor output impedance code dependent means that, there is poor dynamic performance, can you comment about the complexity of the decoder .

Student: I am able to interfere, sir system 2 system.

I mean how many wires must come from the DAC, from the resistor string to the mux.

Student: The average may be in between 2 s.

Student: The changing of weight 2 power.

2 to the power n lines must come from the resistor string into the array, and all these switches must come together at one single node namely the.

Student: Output node.

Output node. So, you can see that this you are picking up a whole lot of parasitic, and therefore this Messy decoding logic and band limiting at the output node, so all these problems basically you are allocated this DAC to places, where you really do not need. You know either much speed or any dynamic performance at all, basic as what it means the nice thing about this is that it is inherently monotonic, and the fact that the decoding logic is messy is also a problem.

So, you use it in the places, where that the logic does not get too Messy, in other words where the resolutions are not very high, and when you are not really interested in dynamic performance. Can you explain I mean can you tell me where you have seen this being used in the past.

Student: In flash A to D convertor.

In a flash A to D converter that is basically what you are doing, the resistor string is also if you think about in a another light, you can think of it as a DAC, where all the outputs are available, simultaneously and you pick which 1 you want, that is what the mux is doing. So, when you had a flash array where all the references were needed at once, this seems like a natural choice, that this resistor string has got all DAC outputs available at once, and all these are going to the comparator, you know offset I mean whatever capacitors which store the reference.

And in that particular design example I discussed where, we needed only one reference at a time, all that you need to do is have an arrangement like this, and this mux figures out which of the taps of the ladder to pick and you know give to the that particular comparator. So, that is basically a DAC in disguise you understand which are resistor string DAC, again because you have a lot of time for auto zeroing and so on, the settling time of the DAC is not really a big issue, and the fact that it is inherently monotonic is advantages.

So, only in such applications you use a resistor string a full resistor string DAC, so in the next class I will discuss other topologies, and see what their limitations are and what 1 can do about it. The fact that mismatches a problem is you know, it is not as if know you magically get away from it, mismatch will always be a problem regardless of what DAC topology you chose. Thank you.