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

Lecture - 51
Binary and Thermometer DACs

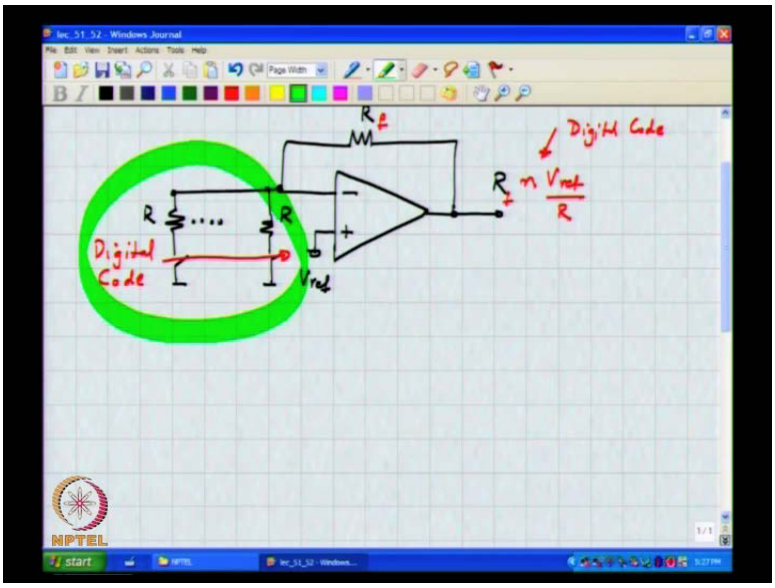
This is VLSI Data Conversion Circuit, lecture 51, in the last class we were looking at data, I mean digital to analog converters and the one of the simplest methods of converting a digital code into an analog quantity is to perhaps use a resistor string. And depending on what the digital code is you select the appropriate tap of the resistor string as the analog output, as we saw yesterday there several problems with this basic idea.

One of course, is that the decoding logic is very complex, further problem is that the output impedance of this DAC keeps varying with the digital code, which means that the output voltage developed, when connected to some load impedance will be a non-linear function of the digital code. Because, even though the settled value is probably, the way it settles to the final value is a function of the digital code, which means that if we are interested in dynamic performance, which is often a very important thing in lot of applications.

For example, communication or things were the entire waveform is of concern not just the settled value, then using such a DAC becomes a problem. On the other hand there are exist many applications were only the settled value is of concern, and classic example is a flash A to D converter, you can think of the resistor string as basically being a, due to a converter of a special kind were multiple outputs are available at the same time, if you want to for example.

So, the next, I mean another way of converting a digital code into an analog output is to not add voltages, but add currents, so to add voltages, what we were doing. We were taking this reference voltage dividing it up into a whole bunch of small voltages, which are represented by the voltage drops across these individual resistors in the resistor string. And to generate the final output we were adding up several voltages, whatever can be done in voltage can also be done in done with current.

(Refer Slide Time: 02:54)

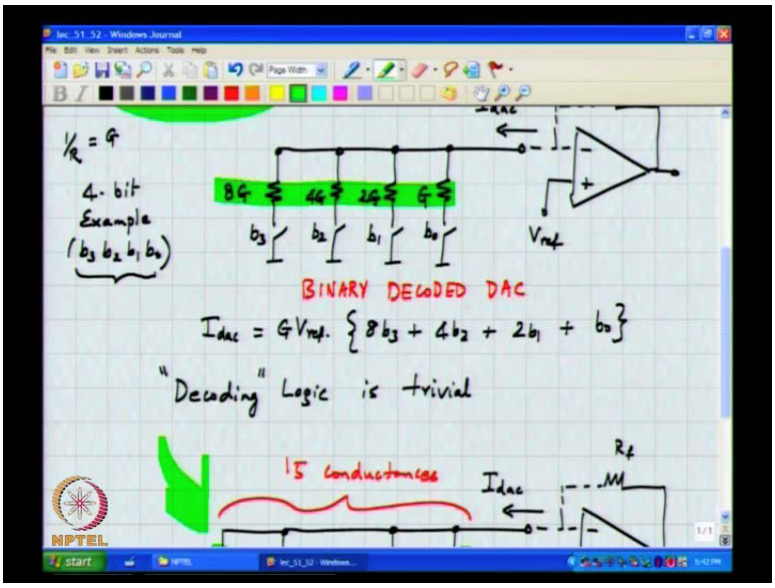


So, for example, if one had a variable current source in some fashion for say let us say this current depended on the digital code, and if you eventually were interested in a voltage this current I, which is a function of code will result in an output voltage. Which is R times I which is a function of the code, where the op amp is simply acting as an ideal current to voltage converter.

Now, several ways of realizing this current exist, you could have a whole bunch of current sources, which were controlled by a digital code, we all know how to make many copies of the same current. We use current mirrors and we generate multiple copies of a same master current, and you can we also know how to turn a current source off. So, depending on the digital code we can turn the requisite number of current sources on or exploiting the fact that this is a virtual ground node, one could have for instance make this V ref.

And we have a whole number of identical resistors R and depending on the digital code, you can either leave R floating or you can ground it, if the resistance is connected to ground a current V ref by R will flow through the resistor. And will result in an output voltage R f times n times V ref by R, this is the digital code or else the decimal representation of the digital code. This is just the dual of the resistor string where there to add voltages, we had many elements in series to add currents you have many elements in parallel.

(Refer Slide Time: 06:26)



Now, one can think of several ways of implementing this current digitally, so one straight forward way would be to say this is my, so called virtual ground of the op amp to the time being. Let us assume that all this infrastructure is there, so as far as we are concerned we are pretty much only interested in the current, that is being pulled out of the virtual ground node, which will flow through the feedback resistor, and gets converted into a voltage at the output of the op amp.

Now, we have work in conceive of many ways if converting the digital code. And you know I mean one can rather what I intend to say was that you can conceive of many ways, in which digital code can modify whatever resistive network we have to generate a current which is you know n times V ref by R. As, a case in point, since we are talking about currents we would always be getting quantities of the form V ref by R and so on, so is of keeping 1 by R, I will just denote 1 by R with the conductance g.

So, for example, let me take a 4 bit example, one conceivable way of converting this code into currents is to have 1 resistor which is 8 g, 1 resistor which is 4 g, 1 which is 2 g, and 1 conductance which is g, the switch on the left is a controlled by b 3, b 2, b 1 and b 0. So, the current I DAC is simply, so V ref times g times.

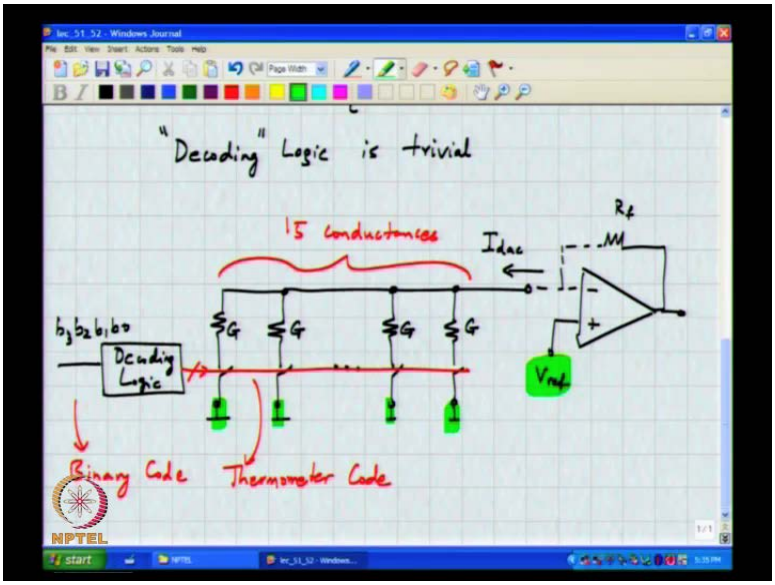
Student: It will be 8 b three.

8 b 3 plus.

Student: 4 b 2.

4 b 2 plus 2 b 1 plus b 0. While, this seems quite trivial one observation that I would like to bring to your attention, is the fact that the logic required to control the switches is actually trivial. There is nothing the 4 lines that come in, or the same lines that go and control the switches, so the decoding logic, so to speak does not exist, on the other hand it is also possible to realize.

(Refer Slide Time: 10:30)



This network using 15 conductance's all having, so when all the switches are turned off no current flows, when all the switches are turned on 15 g times V ref flows, and depending on I mean the number of currents, the amount of current you want you can turn on as many digital switches as you want. So, there are in other words, there are 15 conductance's, and you need some kind of logic, which I will simply call decoding logic, which goes and takes the 4 bit control what that you get of the digital input word.

And generates an appropriate 15 bit signal, the key point is that the number of once in the 15 bit signal must be equivalent to the decimal value of the code that you put it. So, what kind of can you name one code which satisfies that.

Student: ((Refer Time: 13:08)).

one way of doing it is a thermometer code, so this is often this is the binary code as you all know, and one common thing to do is to make this a thermometer code, you

understand. Of course, there is no earthly reason to say this must be V ref, and these must be ground an alternative thing is to.

Student: Where n is ground and.

You move all voltages by minus V ref, in which case the virtual ground becomes real ground, the non inverting terminal becomes 0 volts, and all these become negative the direction of the current, I mean remains the same. On the other hand if these terminal were connected to V ref a positive voltage reference, and the current that will flowing will be in the opposite direction, but all these are fundamentally the same.

So, there are 2 extremes, here we have a case where the decoding logic is trivial, and here we have a case where decoding logic is pretty complex, so to take you know n bits and generate 2 to the n minus 1 control lines, the logic must be pretty complicate at least when compared to the previous case. And you can imagine what will happen when you want to have a DAC which is 10 bits or 12 bits, you need to have a 12 to 2 to the 12 is 4096 of these conductance's, and more importantly 4096 control lines going to these 4096 resistor or elements.

So, the question is I mean is this plain stupid way of doing it or is there something that the latter solution offers which is not there in the form you understand, there are I mean one can also conceive of you know something between these 2 extremes. Let us say you have the MSB's are controlled using this thermometer type decoding, the LSB's could be control by binary decoding.

For example, instead of having b 2 and b 3 control, 8 g and 4 g, I could take b 3 and b 2 and generate 3 control signals, which go on control 3 conductance's of 4 g's, but these the what I have shown here represent the extremes as far as decoding complexity is concerned. One is very simple to decode, and the other 1 is very complex to decode, especially as the resolutions become larger and larger, I hope you see this.

So, one motivation to go from this solution to this solution is to say, well this decoding is complex, but at least I do not have 2 g there, 2 g is a very bad word these days you understand. Something else that you may want to usually in engineering, we have noticed that if you give up something, you usually gain somewhere else hopefully, the question is what do we get, any comments.

Student: Sir which term will decide sigma delta or to be the.

one so.

Student: Sir I do not get.

He’s made a valid point, but so his point is that, if we were doing a delta sigma modulator we know that our flash gives out.

Student: Thermometer.

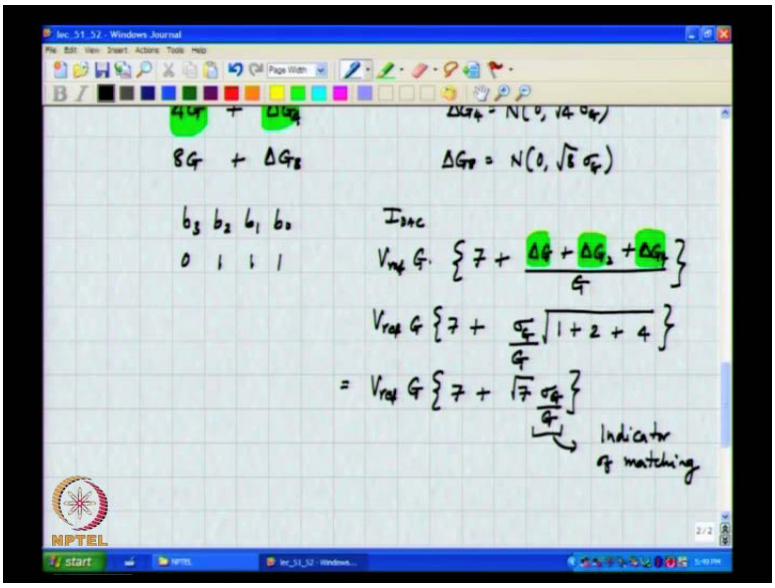
Thermometer code anywhere, so in that case it would make sense to use the ladder DAC, and that is indeed true in a delta sigma modulator, I mean DAC'S like this which have called resistive DAC's are used all the time. So, it takes in a thermometer code and everything is fine, but let us kind of step back a little bit, and wonder if there is more to this then jus it is utility as in a delta sigma loop, where a thermometer code is inherently available.

Student: Previous that is ((Refer Time: 18:30)).

So, if this was exactly twice g, and if this was exactly 4 times g, and if this was exactly 8 times g, and all these nominally identical elements were indeed exactly identical, then there is no difference between the 2 output currents, except that the decoding complexity of 1 is much larger than the decoding complexity of the other. In other words in the absence of device mismatch, the obvious thing to do would be to choose a simplest solution which is the, so called binary decoded DAC or simply the binary DAC.

Now, what point he makes is a very valid and relevant one, we all know that none of these elements will all be within codes exactly identical or exactly the right multiple of g, that we want. They will deviate from each other due to manufacturing tolerances, and by the way how do you expect to get 8 g.

(Refer Slide Time: 21:01)



If you want to get 8 times the conductance, what do you think you will do.

Student: If the input were 8 times.

Take 8 conductance's of g and put them in.

Student: Parallel.

Parallel. So, I mean you cannot even argue that, this is likely to be smaller than this because the number of conductance's appears smaller, that is not the case see there if that struck anyone one of you. The 8 g would you want to get an exact 8 times multiple of g, the sure short way of doing this is to take g and copy and paste 8 times, so you can see that the total number of conductance's used is 15 in both cases.

So, the only real difference in implementation is area of the decoder, now if g is an error other words with mismatch, what is happening every conductance is not really g, but some delta g this delta g happens to be a random quantity characterize by a 0 mean and.

Student: Some

Some standard deviation sigma, we assume is Gaussian and some sigma g, now let us try and understand the performance of the binary decoded DAC with mismatch. So, if an element g has an error which is normal than 0 mean and has a standard deviation of sigma g, can you comment on the error the 2 g element, it is also a random variable. It

should be 0 mean, what can you say about its standard deviation, the root 2 times sigma g you are getting 2 g by taking 2 identical these and putting them in parallel, so the error will be, the variance will be the sum of 2 delta g's.

And because they are random and uncorrelated, the variance will be square root of 2 sigma g, similarly 4 g plus delta g 4, and the delta g 4 will be random variable with 0 mean, and I just make this root 4 sigma g and. So, now, let us look at the step size I mean clearly, if the individual currents change it means that the DAC characteristic will be also moved from it is ideal value, I mean the step size will now differ from the ideal g times V ref that we had.

Earlier, whenever the code changes change by 1, if the elements were all exact and were what we wanted them to be a change in the digital code by 1, should cause a change in the current by g times V ref. Now, because each of the elements is change in some random fashion, it also follows that the step size will change, so let us see what happens when the code was 0 1 1 1. So, when the digital code is 0 followed by all 1's, what happens what is the output current, if there was no mismatch I DAC should be V ref times g, which is a common factor everywhere times 7 plus.

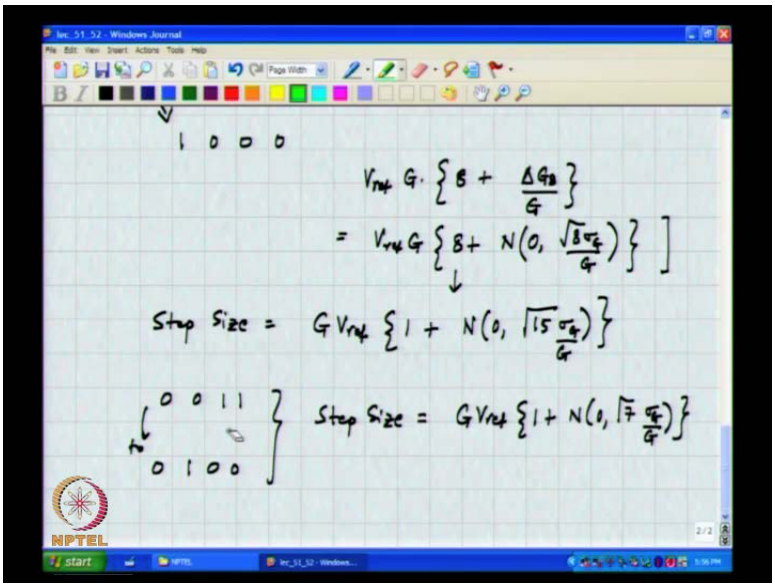
Student: Delta g plus delta g 2 plus delta g 4 whole divided by e plus.

So, if the digital code is 7 the elements which will be excited are this and this, and therefore the errors that will creep in will be delta g plus delta g 2 plus delta g 4 divided by G, and can you comment on whether these errors are correlated. They are not correlate, because they are all coming from different elements, so therefore the DAC current will be V ref times g times 7 plus a random quantity, which is 0 mean. And can you comment on its variance, you do square root of its sigma g by g times square root of.

Student: 1.

1 plus 2 plus 4, which is V ref times g times 7 plus square root of 7 times sigma g by g. Please note that this sigma g by g is a number which is dimensionless, and is an indicator of how well match the conductance's or resistors are, so this is a indicator of matching, now when the code is 1 triple 0, what happens to I DAC.

(Refer Slide Time: 28:20)



Student: If v t to v ref.

Student: V t replace with.

V ref times g times.

Student: 8.

Student: Plus 8 is equal to.

8.

Student: Plus.

Plus.

Delta g 8 by g, so this is V ref times g times, this must be a normal distribution which is 0 and square root of 7 sigma g by g. Now this must, therefore be 8 plus random variable which is 0 mean and square root 8 sigma g by g, now in another words if there was no mismatch if the code changes from 0 followed by all 1's, to 1 followed by.

Student: All 0.

All 0, then the change in a current should be.

Student: V ref

Student: V ref times.

V ref times g, which is definitely true, this transition is called the major code transition it is just Jargon, and it is basically the flipping of the MSB for the first time, as you keep ramping of the codes. Now, can we comment on the step size at the major code transition.

Student: Sir V ref infinity.

The I means this is a random variable, this is another random variable.

Student: It is going to subtract.

Subtract the 2 random variables I mean you can think of this as a gaussian random variable with a mean of 8 and a sigma of square root of 8 times sigma g by g and so on. So, the step size will, therefore be random variable with g times V ref times 1 will be the mean plus 0 times square root of.

Student: 15.

15, and why it is square root of 15.

Student: When the that was.

Student: Only subtracting random variables.

we are subtracting to the random variables is; however, what are we subtracting, we are subtracting delta g plus delta g 2 plus delta g 4 from.

Student: Delta g 8.

Delta g 8 and all these are.

Student: Uncorrelated.

Uncorrelated random variables you understand, which is why the variance is add in the square sense, and therefore you get square root of 15 times sigma g by g. So, this is telling you that the step size at the major code transition will have a variance, which is

not just dependent on the matching, but it is a matching the variance of the individual element multiplied by the number of elements. That are there, which is basically 2 to roughly 2 to the n or the standard deviation is sigma g by g normalized to the LSB times square root of.

Student: 2 power n.

2 power n minus 1 for a large n it is roughly 2 power n. So, why do you think this result makes sense, before we go there let me ask you another question, what about the transition from 0 0 1 1 to 0 1 0 0 can you look at this and tell me what the step size will be. It should be g times V ref times 1 plus n 0 root.

Student: 7.

7, g by g and at this code transition, it will be root 3 and so on, so which is the worst.

Student: 0 forward by all position to 1.

So, the major code transition is the worst as far as the variance of the.

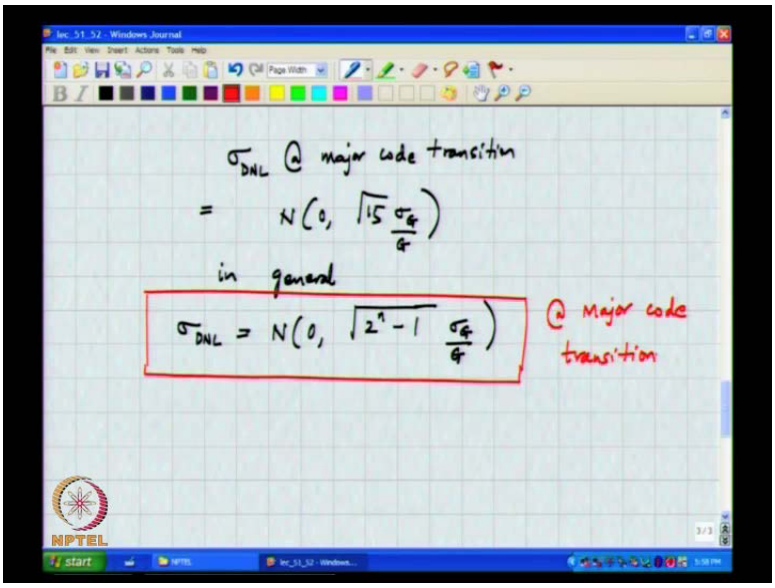
Student: Step.

Step sizes is concerned, variance of the step size is nothing but, what.

Student: DNL.

DNL , so the DNL is nothing but, the ideal step size minus the actual step size minus the ideal step size normalized to the ideal step size.

(Refer Slide Time: 35:45)



So, what is the sigma DNL you get here, at the major code transition.

Student: The operate distribution is 0 mean.

Very good it will be n 0.

Student: It must square root 15.

Square root 15 times sigma g by g, in general it is n 0 square root of.

Student: 2 to the power n.

2 power n minus 1 times sigma g by g. Now, let us put this result aside for while and look at the INL, what is INL, by definition it is the.

What is INL first.

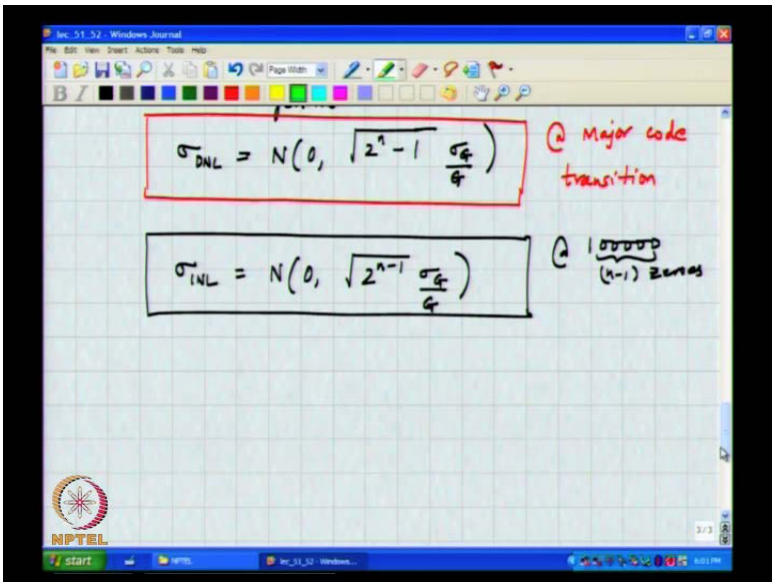
Student: Deviation for the ideal time.

Student: Ideal.

Ideal is a integral of the DNL that is true, but in this particular case much easier to find we know the actual output, we know the ideal output the difference should give you the INL. Of course, because the levels are statistically varying, it means that the INL is also

varying statistically, so what is the INL at the mid code, it is n of at mid code, what is the output.

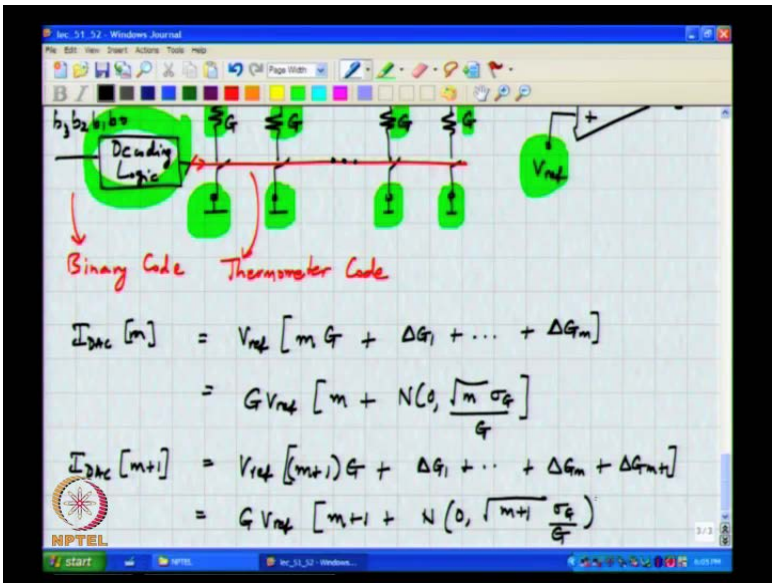
(Refer Slide Time: 38:09)



Let us say this is what we define as mid cod when it is 8, so clearly the ideal output should have been this, so the difference is what corresponds to this part, when we normalize it to the LSB size this will become, n 0 square root of 2 to the n minus 1, 8 which is 16 by 2 times sigma g by g. This is clearly the worst case INL, because for all other codes this number will be small, I mean please note that is the only true when you go and fit the end points, I mean this is after you remove the mean and offset.

So, the first and the last points will be by definition you pin them, so the maximum deviation will occur in the middle. Now, the question is what happens to the DNL and INL, when we use not a binary weighted DAC, but a thermometer decoded DAC, please recall that this way of doing this, because it is based on a thermometer code is called DAC. So, now, let us try and figure out, what the ideal output voltage will be the ideal current will be for code n, it will be n times let me call this for code m.

(Refer Slide Time: 41:34)

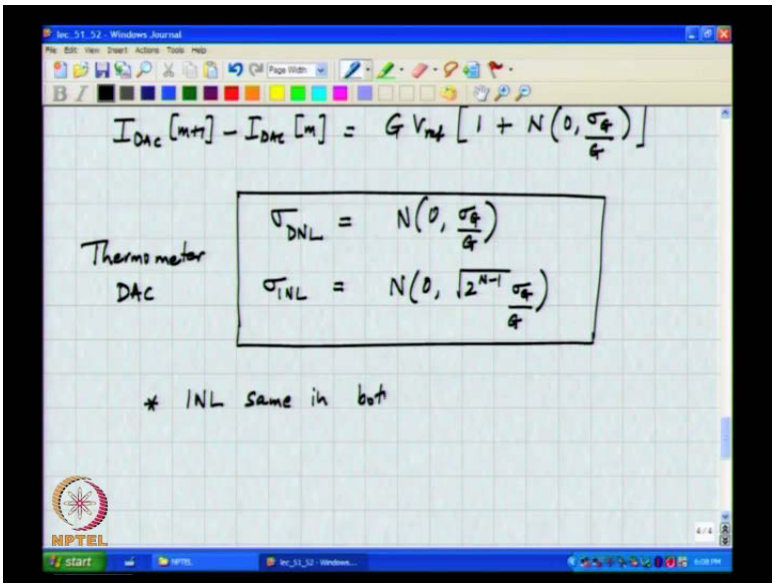


But, a code m the current will be m times g times v ref, but unfortunately every g is has got some random error, and if m conductance's are on, it means that they will be.

Student: M and 1.

M random errors. So, let us call them delta g 1 plus, so as this must be m times V ref times g plus delta g 1 up to delta g m, so I DAC for code n, code m this is clearly a Gaussian random variable m plus n 0 comma square root of m sigma g by g. Now, when the code is increased by 1, this becomes V ref times m plus 1 times g plus delta g 1 plus delta g m plus delta g m plus 1, which is g times V ref times m plus 1 plus n 0 square root of m plus 1 times sigma g by g.

(Refer Slide Time: 44:19)

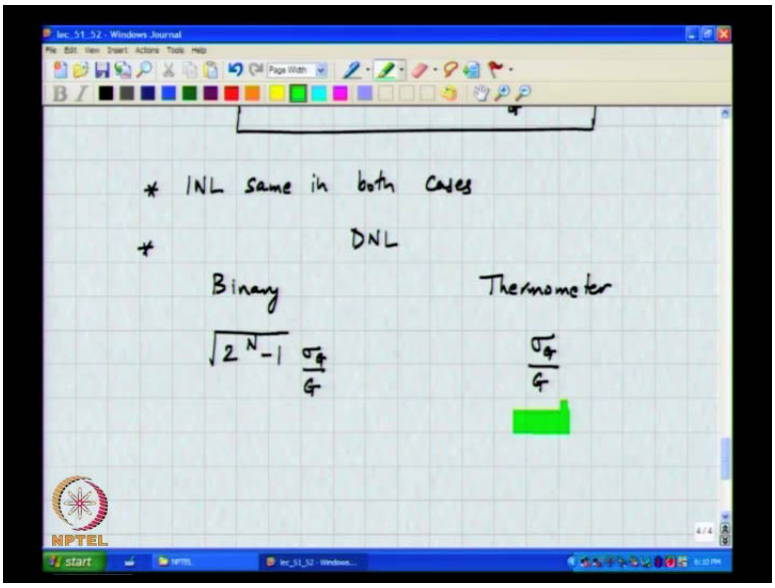


However, what can you say about I DAC of m plus 1 minus I DAC of m is g times V ref, fine times 1 plus n of see that is the catch. It is tempting to conclude that this is 2 n plus 1 square root to 2 n plus 1, but that is not correct, because these random variables are not uncorrelate, please note that this g 1 through g m is the same for both of them. The only difference between the 2 is this extra stuff that you have added, so it is n of 0 comma.

Student: Sigma.

Sigma g by g. So, this means that the sigma of the DNL is a independent of the code and is given by n 0 comma sigma g by g, and the sigma of the INL. It is the same expression that we had, when these expressions are the same thing that we had earlier for a code 8 we had root 8 sigma g by g, in the binary decoding case. Here, it will also be the worst case will become square root of 2 to the n minus 1 into sigma g by g, so the question is of course, this all comes out of the math's, but is there some intuition to this, we see that INL same in both cases, but DNL in the binary thermometer cases.

(Refer Slide Time: 47:09)



It is binary case it is square root of 2 to the n minus 1 times sigma g by g, in this case it is simply sigma g by g, so why do you think, I means which is larger. And this that basically means that the DNL, you get when use a binary decoded DAC will be a hell of lot more at the major code transition. Then the DNL you get with a thermometer DAC is this clear, I mean that is what the math is telling us now let us try and figure out why do you think this make sense.

Student: In fact, using the resistors for I mean a mutual DAC we are using that same set of resistors and 1 extra.

Yes correct.

Student: For x plus, but in binary DAC you are using completely different sets of resistors.

So, the key point is to note that, in a thermometer DAC to change 1 step right what you are doing you adding.

Student: 1 word.

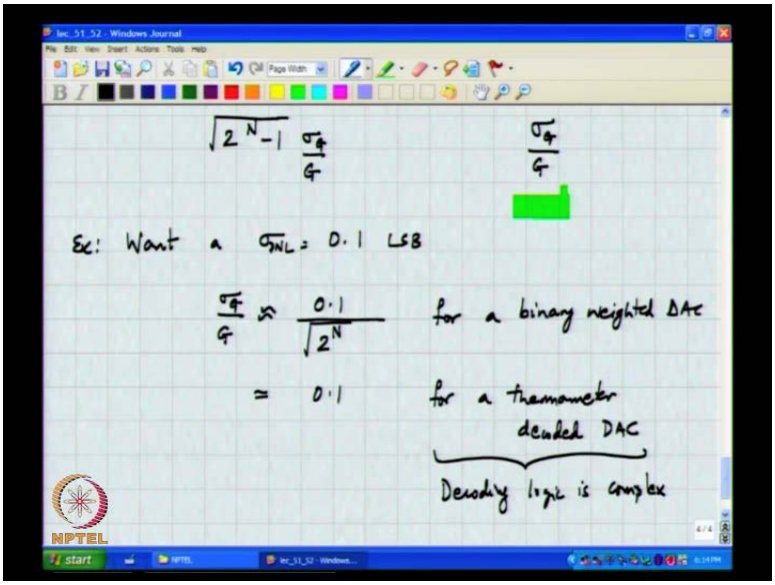
1 extra step, where that step is the same regardless of the code nominally, so you are only worrying about the variance of that particular step that you are adding. On the other hand in a binary DAC when we make the major code transition; however, you getting this

current of g times V ref, we are generating the small current which is the LSB of g times V ref by subtracting to large numbers.

Earlier, 7 different current sources were on, and what is happening when we do a major code transition is that the 7 current sources which are on would been turned off. And a new 8 times LSB is coming on, and these errors are completely uncorrelated, because they are happening in they are coming from different unit elements.

So, when we subtract 2 large numbers to generate a small number, I mean this is something that we all know, if we get a small number which is the subtraction of 2 large numbers, we know that small quantity is proven to be very erroneous. Because, there is error here, there is error there, this large number does not know, that this I mean what this other large numbers is doing when you subtract of course, on the average you get the right value, but they will be huge variations, that is what this is telling. So, if the DNL is spec which it will often be.

(Refer Slide Time: 50:44)



So, let us say you want a sigma DNL of say 0.25 LSB, this is just let me say 0.1 LSB, so that 3 sigma DNL is 0.3 LSB, if this is what you want as a designer. It means that you must go and choose sigma g by g to be equal to 0.1 by square root of 2 to the n minus 1, which is roughly square root of 2 to the n. For a binary DAC and should be equal to 0.1 for a thermometer decoder, so what can you say about the INL.

Student: It remains same.

Student: Same.

INL remains the same in both cases, so but whatever we paying for in terms of a thermometer could decoded DAC.

Student: decoding logic.

The decoding logic is complex, whereas for a binary weighted DAC, binary decoded binary weighted is also another common term, you get the logic is lot simple. So, next class we will continue with more discussion on these ways, and you know that you know these are 2 extremes, so in practice when you have a high resolution DAC. The best path is probably the middle path, which is neither all binary nor all thermometer right some something in the middle, so we will continue after a break.

Thanks.