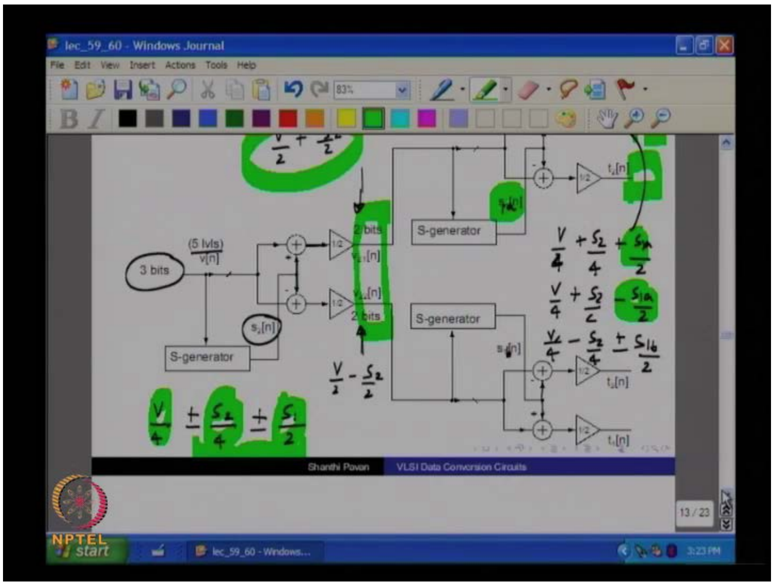
VLSI Data Conversion Circuits
Prof. Shanthi Pavan
Department of Electrical Engineering
Indian Institute of Technology, Madras

Lecture - 60
Dynamic Element Matching – 2

(Refer Slide Time: 00:29)



This is VLSI data conversion circuits lecture sixty in the last class we were looking a prototype DAC, where the noise due to mismatch is shaped outside the signal band and you can leave it. When you can the basic idea is to divide and conquer? So, you take for instance a 3 bit DAC with 5 levels and split it into 2 sequences, which occupy one bit less.

So, there are fewer levels and 1 of them is related to the other through this sequence s. So, the output at this layer is nothing, but v by 2 plus s by 2 and v by 2 minus s by 2 since this happens 2 layers away from the last layer I call this s 2 now since we have 2 bit sequences each of these can be split into 2 1 bit sequences and, finally what you have is v 2 v by 2 by 2, which is v by 4 plus s 2 by 4 plus s 1 a I will call this s 1 a s 1 a by 2 and v by 4 plus s 2 by 4 minus s 1 a by 2 correct. This is the same and the only difference is this s 1 a by 2 and, similarly here you will get v by 4 plus s 1 v by 4 plus minus s 2 by 4 plus minus s 1 b by 2.

So, basically the thermometer drive wave forms will be v by 4 plus minus s 2 by 4 plus minus s 1 by 2. Where you know? Plus happens to be s 1 a, and a s 1 b respectively. So, clearly we see that each thermometer drive wave form is linearly related to the input and has got an addition these sequences have a spectrum, which we have designed to be shaped. So, if there was no mismatch in these unit elements, which the thermometer drive sequences will be driving. What do you? Can I expect to the combined wave form.

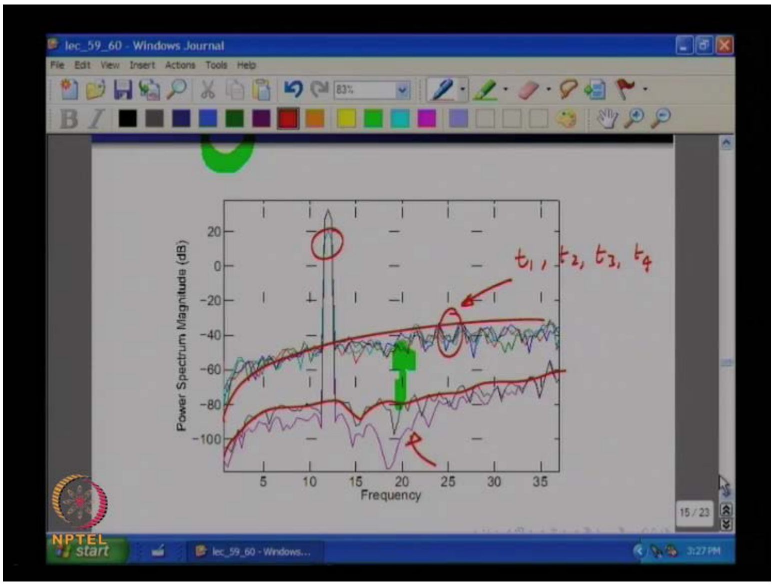
It will be exactly the same spectrum as v, because at every layer of hierarchy we have insured that the sum of the split sequences is exactly the same as the input, which means the sum of all the thermometer drive wave forms right or the drive sequences at any instant of time will be exactly the same as v at that instant of time does it make sense, now if there is mismatch what do you expect to happen.

There will be some residuals of s 1 and s 2 coming out and the shape of these sequences the spectral shape of the sequences will appear in the final output spectrum in the if we assume first order noise shaping then for s 1 and s 2 then when there is DAC element mismatch you will find that the mismatch noise, which was white when we randomized should now have a first order noise shape all right, and this should be a huge improvement over all pure randomization does it make sense.

(Refer Slide Time: 05:26)

So, let us before I show you the simulation results this is the spectrum of v, and I have deliberately you know offset these 2 sequences v 1 and v 2 1 is in orange and 1 is in green, and the sum of these I mean this is actually the 0 for these 2 sequences I just off set them for visibility on the graph the sum of these 2 sequences v 1 plus v 2 is v and how many levels are there in v. These are five level sequences. How many levels? Are there in v 1 and v 2. Three level sequences correct. So, how many bits are required to represent this. Three bits what about this.

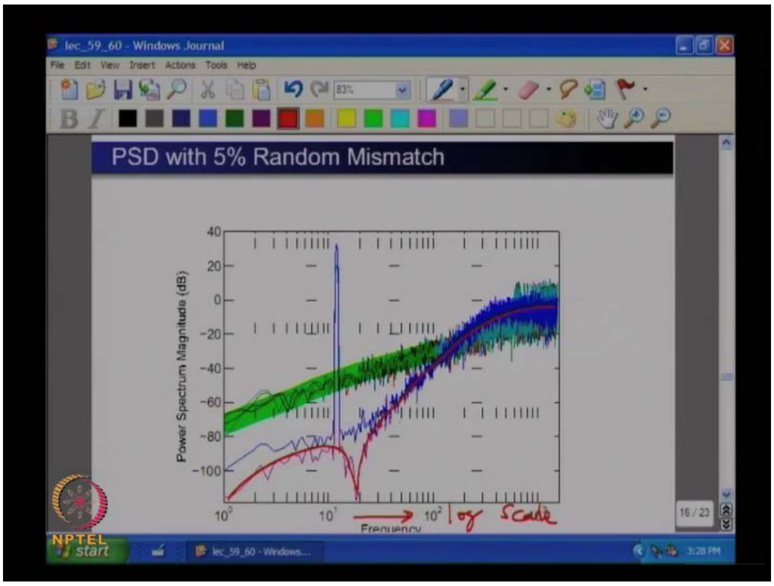
(Refer Slide Time: 06:52)



And, now you can go and decompose v 1 and v 2 into 2 single bit sequences, and these are the p s d's of t 1 t 2 and t 3 and t 4, which are the p s d's of the individual. Drive wave forms of the unit elements and as usual we can see that the signal component in each 1 of them is the same correct, and you can see this shaped sequence and this corresponds to that s 2 by 4 plus s 1 by 2 or vice or whatever the important thing is to note that this is not.

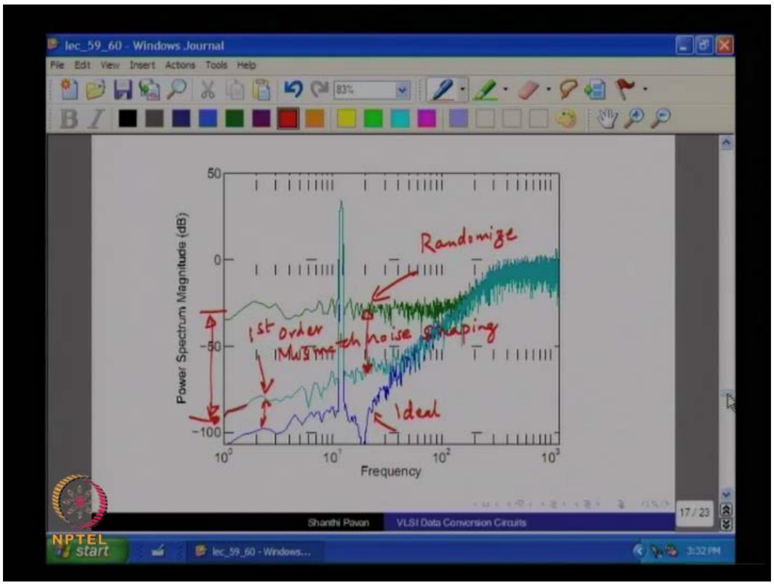
A noise shaped spectrum first organized shaped spectrum this is the spectrum when there is no mismatch and that corresponds to same n t f and stuffed that we have been using for all our simulations correct ok, and clearly when there is mismatch the cancellation of these shape noise sequences that shape noise part is not exact, where left with a residual right for 1 percent random mismatch you see that the you are little above the ideal s n r that or ideal s q n r that you would have got you understand. So, now let us see this on a log scale and with 1 percent mismatch the now the noise due to mismatch is. So, close to the quantization noise that I mean you are not able to see the first order noise shaping of the mismatch noise itself ok all right. So, if you plot this on a log scale with more mismatch. So, that the mismatch noise is dominates this is 5 percent mismatch and this is a log scale as evident from the graph.

(Refer Slide Time: 09:11)



These are the spectra of the individual thermometer drive wave forms and as you can see on a log scale they must go as twenty d b per decay right at which is definitely happening and when there is mismatch you see that the residual is also going at twenty d b per decay and this makes sense right ok, and the ideal 1 is this .part and the spectra of the mismatch shaped DAC and the ideal DAC coincide at high frequency on only at a low frequencies, where they were supposed to cancel out right, and very small residue now what is 1 obvious comparison that we would like to do now we have seen 2 techniques, where distortion is removed right one is pure randomization and other one is this noise shaping right on obvious thing is to say is to see how we are doing in this case compared to the other one.

(Refer Slide Time: 11:00)



With 5 percent random mismatch you can see that this is the ideal all right this is with randomization all right, and this is with first order mismatch noise shape right and you can see that you have gained, significantly right if you assume that this is the band edge and most of it is getting contributed here you can see you have gained a good forty fifty d b right and this 5 percent mismatch happens to be you know a really a bad example as far as mismatch is concerned I mean just, because you have something which shapes the mismatch noise out it does not mean that you can go and build a deliberately go and build a lousy DAC.

So, it turns out that I mean if you do within codes use standard best practices for a layout right it is not uncommon to get matching in the order of half a percent or. So, right sometimes capacitors can match to if you do a good job you can even match to like point 1 2 point o 5 percent right, which is at their matching at the ten to eleven bit level. So, even if this 5 percent mismatch, and your this much above the quantization noise floor if the mismatch became half a percent rather than 5 percent what do you think will happen.

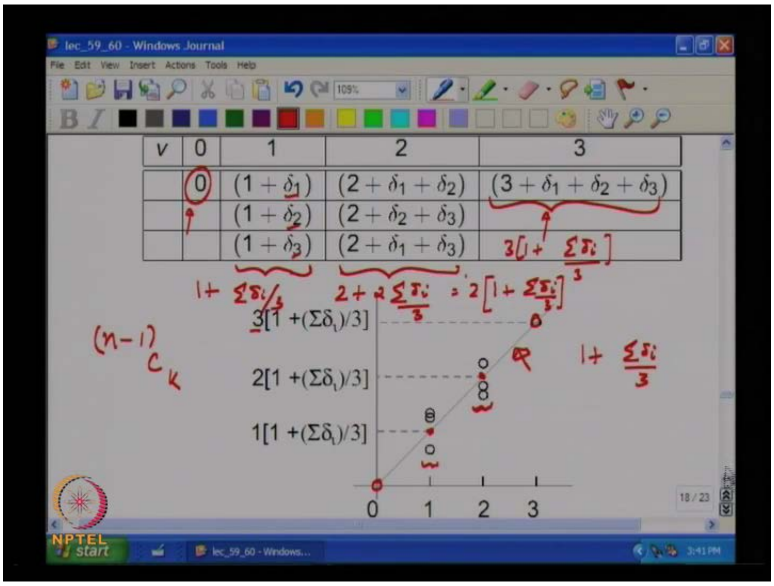
Student: Pardon.

What will happen? I thought we had discussed this last time around. It will go it will go down. It will go down by in half right. Power is go down by ten d b right by twenty d b right, because it is a square and then power right. So, so that basically means that if you have a half a percent distortion you probably would not be able to see I mean with mismatch with first order mismatch noise shaping at the spectrum you would see pretty much. Same as the ideal spectrum does it make sense ok. So, 1 thing that you know 1 could ask is hey what is. So, special about first order noise shaping why not.

Shape it with any higher orders I mean that is definitely possible except that the I mean this thing becomes a little more complicated right and. So, on and actually most of the time right at the mismatches are always said is small enough right let us say half a percent is you know conservatively speaking you know an all right number to choose for mismatch in the unit elements in, which case you know at these levels like hundred plus d b levels you still do not see any degradation due to once you have a first order noise shape, because once you have first ordered mismatch noise shape you are almost.

The theoretical you know s q n r level. So, then you might say what is the point in trying to make a second order noise shape version where? It is true that the mismatched noise will be even smaller, but you dominated by quantization noise anywhere. So, I bother right. So, a lot of times people just stand up stopping with first order mismatch noise shape you understand. So, this is I mean. So, this line of investigation right has been you know is responsible for tree shaped or tree structured mismatch noise all right, and you can see that the logic that there is a logical progression between what we wanted to do, and what we eventually did correct a historically of course, and the you know people thought completely differently and so you found that o I mean that line of thought gives rise to whole bunch of mismatch noise shaped architectures, where the benefits due to mismatch noise shaping are not as evident as they are here right here we knew what we have doing we implemented exactly that, and the fact that we should expect mismatch noise shaping the fact that we should expect first order residual noise is all very apparent right.

(Refer Slide Time: 16:55)



But, the original part process behind mismatch noise shaping is somewhat different I will again illustrate this with our 4 level examples. So, there are three unit elements. So, I mean when the levels of the DAC please note that again as I said this line of thought is somewhat more ad hoc than then the we the clean set of reasoning that leads to the tree structured mismatch shaped DAC all right.

In many of these techniques also happened to be first order will give you mismatch noise, which is shaped to first order right, but that is not evident from the discussion or unless you know you think about very carefully it is not very obvious that the mismatch noise shaped I mean the shape noise is actually first order all right, and many of these have are if you kind of do a search with you know dynamic element matching you know the literature for the last for the last twenty years you will find schemes, and several of these techniques, which are you know kind of based on the intuition that I will describe are what do you call not as nice and rigorous as the tree shaped structure and, but I will explain the basic intuition as I go along all right.

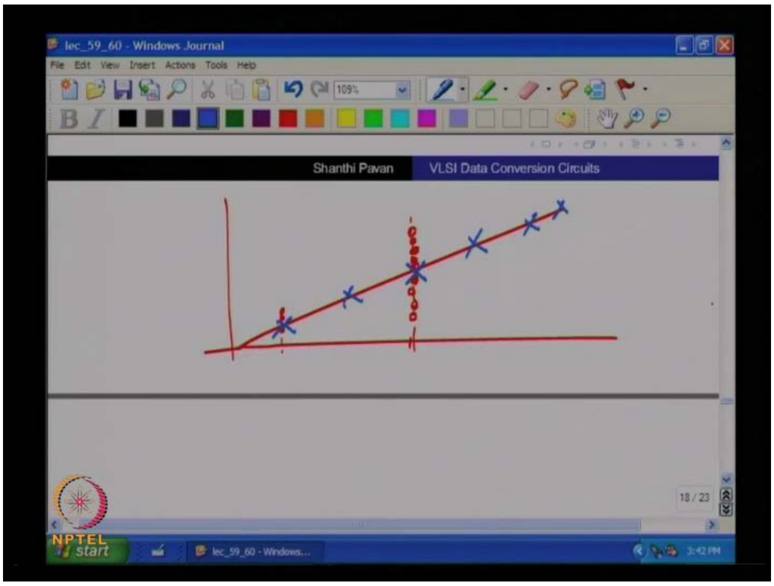
So, if you consider a 4 level DAC with 3 unit elements you have when the code is either 0 or 3 there is only 1 way of choosing the elements when the code is 0 all elements are off when the code is 3 all the 3 are on. So, whether it is 0 or 3 there are if there is no freedom of choice if the errors in each 1 of these elements are delta 1 delta 2 and delta 3 then you the full scale s 3 plus delta 1 plus delta 2 plus delta 3. The lowest point is zero.

However, in the middle for example, if the input code is 1 then there are 3 ways of choosing 1 element right, and each 1 of these ways results in a different error. Now, there are 3 ways of choosing two elements also.

And, these are shown here. So, each of the circles represents 1 way of choosing elements such that you get a nominal value of 2 now the interesting thing is that if you look at the center of gravity of each of these levels what is the central gravity of this there is only 1 level. So, that will be 3 times 1 plus delta 1 plus delta 2 plus delta 3 by three. So, this is I have just written this as 3 times 1 plus sigma delta I divided by 3 now what is the center of gravity of these 3 levels is nothing but. Take all these 3 add them up together, and divide it by three and that will be 2 plus sigma or 2 sigma delta I divided by 3, which can be written as twice 1 plus sigma times delta I by 3 is not it. So, and similarly for this it will be 1 plus sigma delta I by three.

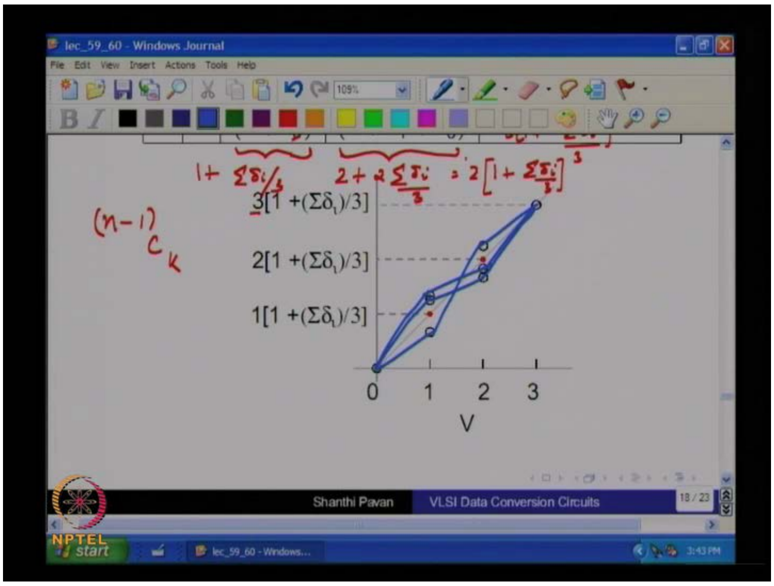
So, and the center of gravity for 0 is zero. So, if I draw a straight line as shown here are you able to see that dull grey line if that has a slope of 1 plus sigma delta I by 3 right. The center of gravity of each of these codes lies on a straight line I have chosen I have shown this for a 4 level DAC. But, it is 2 for an arbitrary number of levels if you have an n level DAC then there are n minus 1 unit elements, and the number of ways of choosing a code k is n minus 1 choose k there will be. So, many I mean so, in general for a large number of levels.

(Refer Slide Time: 22:53)



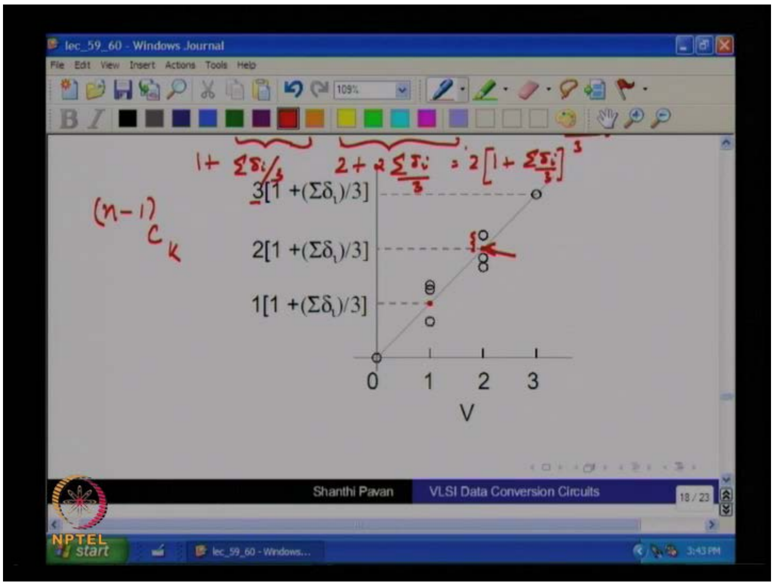
You will basically, find that there will be I mean you can draw hypothetically straight line and for each code there will be a whole number of ways doing this right, but the center of gravity of all these will lay on a straight. So, based on this observation right a whole bunch of you know schemes came up. So, to give you some intuition if a. So, ideally I mean if you use thermometer type decoder your DAC would have the same element every time.

(Refer Slide Time: 23:53)



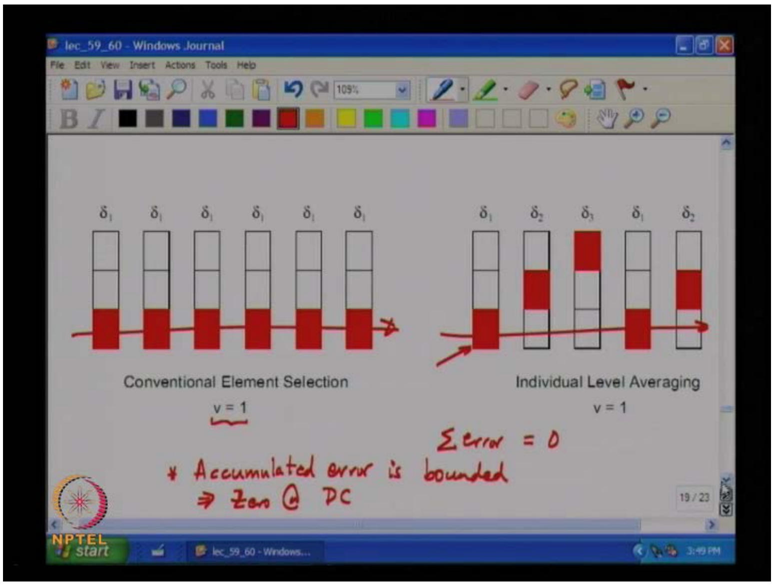
So, the characteristic of the DAC would probably be some non-linear thing like that correct. So, then the obvious thing is to say on the average I want a straight line. So, sometimes I will choose this sometimes I will change my choices of unit elements, and do this keep changing this often enough and therefore, on the average I must get the straight line.

(Refer Slide Time: 24:53)



So, this is pond a whole bunch of techniques the first 1 based on this observation let me also point out another way of saying the same thing the center of gravity all these codes is this red dot shown here correct. So, if I look at the deviations of this red dot the sum of all the deviations must be definition that is the center of gravity is not it right. So, the sum of all the deviations of a particular code from that hypothetical straight line right, which passes through the center of gravity of all these code choices is zero and that is true for any code.

(Refer Slide Time: 25:45)



So, based on that intuition the first technique 1 of the first techniques that came up was what was called individual level averaging and the idea is the following conventionally if we have v equal to 1 only the lowest element in the thermometer decoder will be 1 or will be driven to one. So, you will be using this always now with individual level averaging what you say is that the first time I get a 1 I will choose the first 1 the next time I get it I will choose the second 1 the third time I get it I will choose the third unit element, and the fourth time I will come back to the first and keep repeating right.

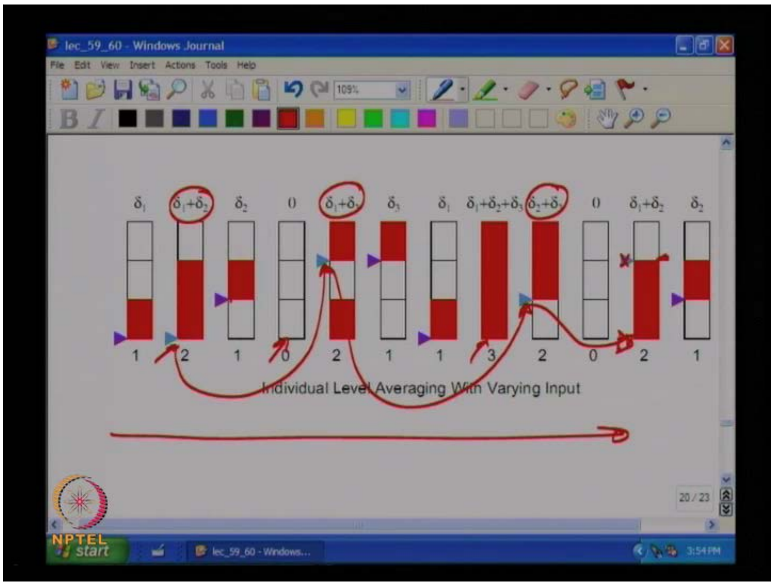
So, we can see that in 3 clock cycles the error. So, the absolute error is this, but the deviation from the hypothetical straight line goes to zero once you have once you go through. All these things ok and. So, on right and though it is not apparent until its pointed out you can see that the thermometer drive wave form will be will have a mean which is V by n level minus 1, and in general if you have slowly wearing of t v of n you can expect the mean value of each of these thermometer drive wave forms to also be V by n level minus 1 ok. It is not very apparent that mismatches is noise shaped right since I mean 1 thing you can say is that if you think of the error sequence the error goes to 0 after or sorry they the accumulated error right. So, sigma error goes to 0 after every 3 cycles in this exam right. Now, if the accumulated error goes to 0 after 3 cycles in general you can say that regardless of the if you go on if you look at the accumulated error after a lot of time right this accumulated error is always bounded yes, because after every 3 cycles it is going to 0.

So, wherever you stop this accumulated error is bounded does it make sense or this accumulator is finite. So, if the accumulated error is finite then what can you say about its low frequency content. If the accumulated error is bounded in other words if you have wave form you integrate it, and its output voltage is constant the output of the integrator is constant. What can you say for the content at low frequency.

Student: DC value must be zero correct does it make sense.

Other way I mean if there was a non-zero d c content you integrate it will blow up right. So, if you have a sequence and if you accumulate it if is still remains bounded correct, which means in the worst case you can only have a d c content after accumulation it must follow that it has a notch at d c right. If it has 0 at d c then it follows that at low frequency it is probably small does it make sense. So, that is the intuition for. First order noise shape right. So, the accumulated error is bounded, which means 0 at d c.

(Refer Slide Time: 30:59)



So, when the input is varying what is done is the following. So, I hope I would see 2 colors for the pointers please, now just focus your attention on the one is in the input all right. So, when the input is 1 we start off with we have a pointer which starts off at the lowest element and this pointer is colored here in violet the next time I get a 1 right this pointer you start from the pointer is simply accumulated version of where the pointer was earlier plus 1 the third time and then the pointer must then move to 2 and then the next time you get a 1 the pointer must move module o 3 and then the pointer moves right.

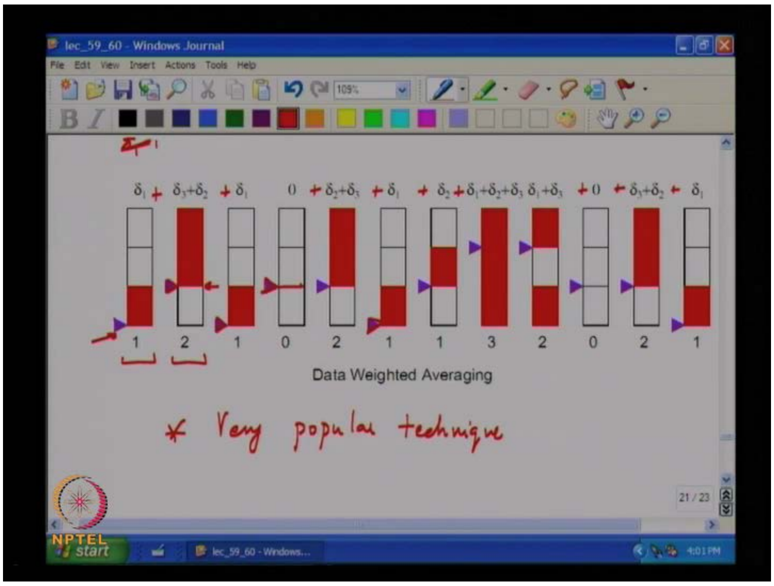
So, for each input level you have a pointer right, and that pointer keeps you keep starting from where you left off for that particular level at that individual level right, and I mean in other words it is like this there is no difference between. So, if we just look at those instants of time, where v equal to 1 if I picked out only those features and put them side by side this is how it will look correct and as we already saw the accumulated error here is is bounded, which means that it has a 0 at D c. Now, if the same thing is done for all other individual levels then the same thing will happen to the errors contributed by. Those sequences also.

So, for example, for the input level of 2 you start off with 0 the next time around the pointer moves to 2 then the next time you get 2 which is here 2 plus 2 is 4 four modulo 3 is one. So, the pointer moves there the next time you get a 2 you start you start here, actually this is a mistake you start here and the next time around the pointer will be the pointer for 2 will start later does it make sense, and just like and look at the error here it is delta 1 plus delta 2 the error here is been delta 1 plus delta 3 and the error here will be delta 2 plus delta three.

So, when you add these 3 sequences it will become 0 I mean, whenever you hit delta 1 plus delta 2 plus delta 3 times 2 right you are done correct. So, if you take a sequence which is just for these two’s that must also have a notch at d c correct for 0 and 3 there is no need to have pointers, because there is only one way of choosing them correct. So, if you look at the accumulated error sequence since each 1 of these sequences each 1 these accumulated error sequences bounded it follows that if you find the accumulated error sequence for this when the input is varying that will also be bounded yes which means that at low frequency you will have at d c a notch at low frequency it will go as 1 minus z inverse.

So, this I mean this still does not mean that there is no harmonic distortion right because it is not clear at least from the discussion we had that these thermometer drive wave forms are linear functions of the input I mean it is true that the mean of the thermometer drive wave form is same as the same as the input, but them and that does not mean anything right, you could have you could also have some distortion components and that could then show up as tones in the output.

(Refer Slide Time: 36:49)



Now, w a very popular algorithm, which is variant of which is slightly different is what is called data weighted averaging and the idea is this. So, the pointers for individual levels are done away with. So, let us assume that this is the sequence the pointer starts at 0 or at the lowest level then the input sequence is 1 the pointer moves here the input sequences 2 and. So, you start the thermometer from here, and go there and the pointer comes back to the lowest level, because it is you know modulo 3 right if you start from 1 actually it should be modulo 2 0 1 2 ad then comes back. So, then it is 1 it goes here then the input is zero. So, you stay there then it becomes two.

So, the pointer moves to 0 again and. So, on and. So, on and again as you can see what happens the accumulated error is delta 1 right plus delta tow plus delta 3 plus delta 1 right and plus plus plus plus and so on alright, and please note that you want to look at the deviation of this from you must assume that what you are interested in is not deviation from the ideas line with the slope of one, but from the line with a slope of 1 plus delta 1 plus delta 2 plus delta 3 by 3 right.

So, you should subtract you know. So, basically you are looking at deviations from the center of gravity all right. So, once you do that you will find that this accumulated error is also bounded which means this also must lead to first order shaping of the noise does it make sense. So, and this is a very popular algorithm used in in many many practical delta sigma modulators 1 of the selling point of this technique is that the implementation is actually quite straight forward. So, what do you need to do is you need to have pointer and you need to keep accumulating and module owing the result and you just keep starting the thermometer the next time from where the pointer stops.

So, you have a pointer thermometer code starts from there and you keep counting v number of elements, and if you over flow you start again form the bottom right 1 thing you must understand is that again for every code the number of unit elements, which are on is exactly equal to v right if v of n is the code for the number of unit elements on is v of n regardless of what dac's all the dac's we've seen whether is a randomization whether it is a tree based p m where it is individual averaging or whether it is data weighted averaging all right in the absence of mismatch all of them will perform the same way right.

So, this is a me just add this is a very popular technique all right again if you look at each of these thermometer drive wave forms they will contribute equally to the total output does it make sense yes no all right and and it turns out that you can show rigorously that if you want a unit element type DAC to be insensitive to or less sensitive to mismatch then each 1 of these drive wave forms must be a linear function of the input plus some some shape noise type component so.

So, I mean. So, far we have said that is true then it will be insensitive to mismatch right, but you can also show rigorously that the opposite is true which is if a DAC is insensitive to mismatch then. Each one of these thermometer drive wave forms will be linear functions of the input plus.

Student: Shape.

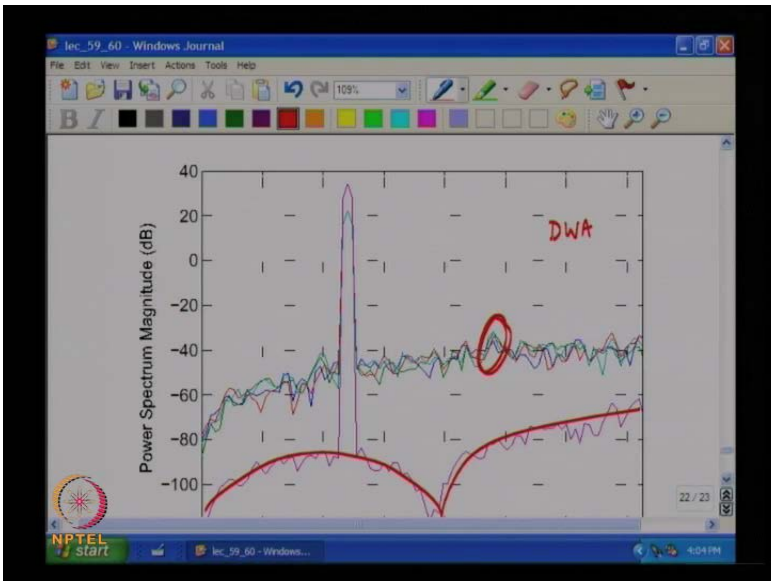
Shape noise right. So, regardless of you know whatever you do to come up with the final DAC right if it is indeed doing mismatch noise shaping without adding distortion and. So, on it will turn out that each 1 of these thermometer drive wave forms is what do you call a linear function input plus.

Plus plus some shape noise with I mean with d w I you can actually in a work through this pointer business and and show that that is indeed the case right it is a very kind of a messy thing, but that is what. So, so lot of these things were not apparent when when d w I came out for the first time right I mean. So, the the realization that that each thermometer drive wave form must be linear must be linear function of the input plus shape noise I think took about at least may be about 6 seven years after the after all these dynamic elements matching technique started coming up. So, you can see that the way is been presented in this class is not the chronological way in which things happened right.

So, you'll find unfortunately that a lot of the literature is is is pretty much is a new technique to do dynamic element matching and you look at it is not immediately clear why you know it is you know why it works or why it is any better than you know prior art chronologically the first technique that was proposed randomization right were people I mean saw that distortion goes away, but the noise floor rises and then these mismatched shaped technique started coming out and as I said the first family of techniques were all based on this observation that if there are many ways choosing elements to get the same output right to get a output of 1 you can choose you know are particular example unity elements in 3 ways and. So, on right and they observed that they you know the average of all these the centers of gravity will all lie on a straight line.

So, lot of the techniques kind of exploited that observation right its only I think in about the mid ninety's around ninety 3 or ninety 4 that people realize that you must you can realize this mismatch insensitive type performance or mismatch shaped performance by making each one of these drive wave forms a linear function of the input plus some shape noise, and that naturally lends to the tree type structure right there are other equivalent ways of looking at the problem right and you can find that in Shreyas text book.

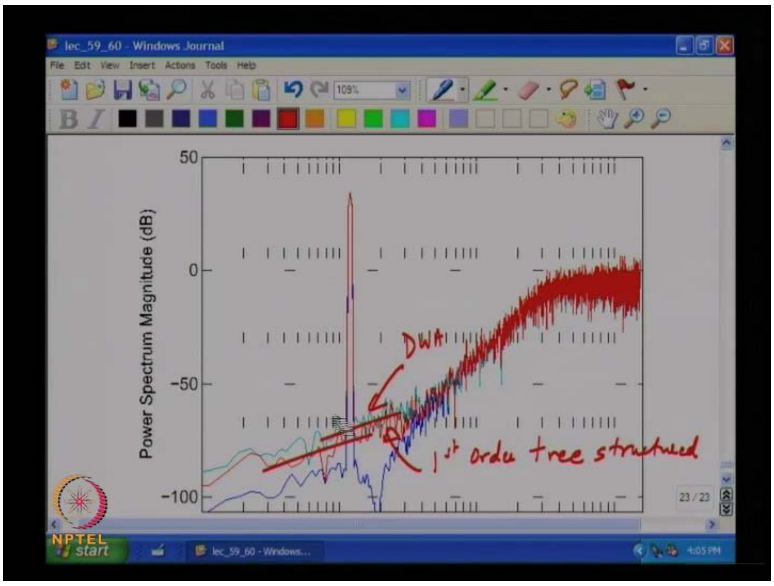
(Refer Slide Time: 45:49)



But this is bottom head to say. So, finally, to just look at the drive wave forms of the individual elements in the d w a algorithm, you see that this s the ideal spectrum, and these are the spectra of the. Individual drive wave forms and it is indeed.

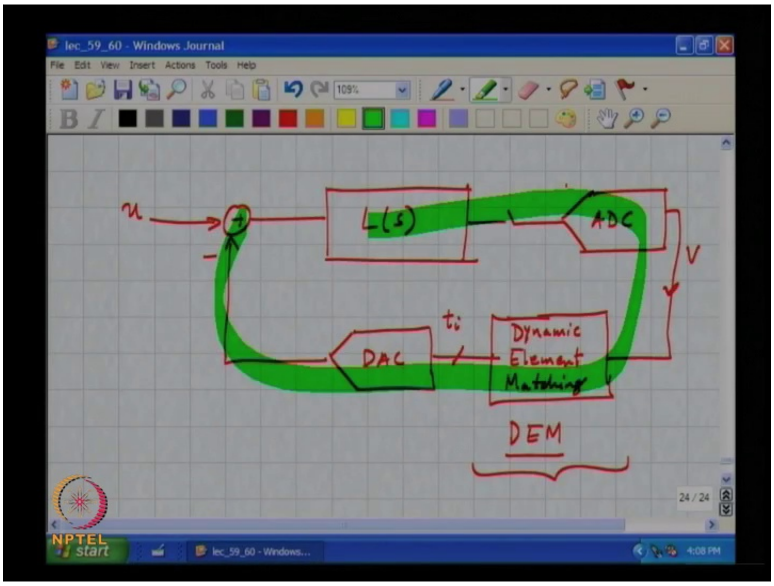
Student: First order.

(Refer Slide Time: 46:21)



First order noise shaped and if you compare this is 1 of them spectra of d w a and the other 1 is first order tree structured data weighted I mean mismatch shaped mismatch shaping of of noise sorry, noise shaping of mismatch all right you'll see that the kind of are the same first order slope they are slightly different here simply because the absolute numbers chosen for the mismatch were different in the 2 simulations. So, the some I mean these 2 are not bang on lying on each other right.

(Refer Slide Time: 47:21)



So, the bottom line therefore is that in a sigma delta modulator if you want to realize it as a multi bit structure, you want to have a loop filter which is either a continuous or discrete time you have a multi bit flash a d c right. And before you go into the DAC you have to get into this block logic which takes the code from the a d c figures out what to do in order to make these thermometer drive wave forms linear functions of v plus some shape noise for. So, this element this is called dynamic element matching in the sense that the matching is not intrinsic right you are getting matching dynamically by doing stuff to the thermometer drive bits.

So, this is often abbreviated as d e m now this is combinational logic which therefore, has the penalty of… Pardon. So, this I mean logic will add.

Student: Delay.

Delay correct. So, there will be I mean. So, d e m will cause axis loop delay all right that might be small when you are operating at low sampling rates, but can become a problem if you are trying to reach a realize very high speeds, and the delay of this may become a significant fraction of the closed cycle does it make sense. Thank you.