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

Lecture - 49
Flash ADC in a Delta Sigma Loop

This is VLSI Data Conversion Circuits lecture 49, in the last class we saw various aspects that govern the timing of flash A to D converter and how one would arrange, the times of the sample and hold and the pre amp and the latch, in order to be able to maximize the time available for each particular, sub block. And we also saw a case study where all these concepts have been put together to realize, a flash converter and you saw that the pre amp is pretty much the same kind that we discussed in class, the latch is the same that we discussed in class and so on.

The only minor trick if you will was that one could actually remove, a comparator from the array and not worry about it too much. Because, one could use the bubble correction logic to mask, the random decision that the comparator gives this way. One finds that one can continue to out of zero all the comparators in the background without affecting, the conversion SNR too much.

Because, only when the comparator which is close to the ideal thermometer transition point from all one is to all zeroes, that when choosing a pulling a comparator out of the array makes a difference and we saw by calculation, that it is a only about ADB or ADB and a half. And especially when you duty cycle this over a long period of time, the effective loss in SNR is a very small number.

The other trick was that any static distortion in the sample and hold can be corrected, by simply pre distorting the references in the same manner and this way you know you get cancellation of the distortion. Thereby improving the or increasing the potential swing that is possible and once you are able to efficiently utilize, the maximum peak to peak differential swing you are; I mean you are in a position where you can within quotes increase, they random offset that you can tolerate which finally, boils down to power reduction.

And this is a general principle in analogue, it you always want to operate with signal swings, which are as large as possible of course trying to make signal swings very, very

large will fundamentally mean that you have to worry about distortion right. So, we have seen this in many contexts, I do not know if you remember, but in the very beginning of this course we were talking about a sampling switch and a capacitor.

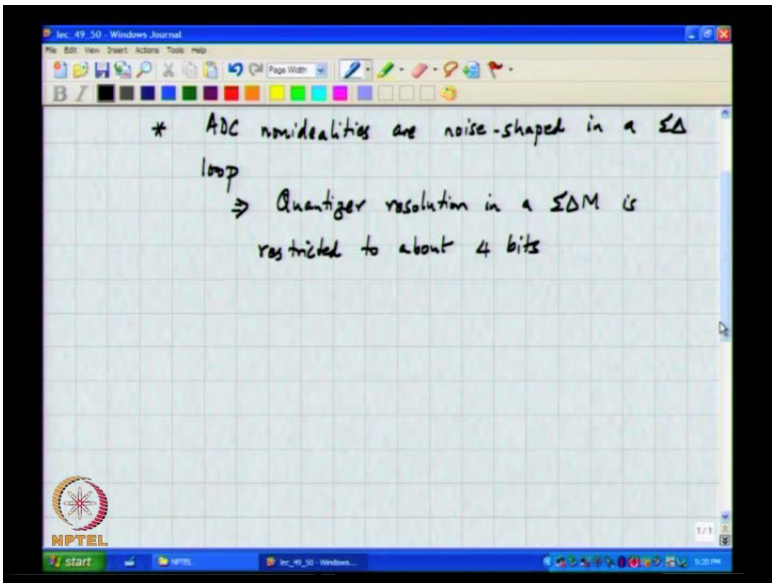
And we said that, the distortion and in the sample and held wave form is proportional to signal amplitude square like it always is, and then one can take the easy way out and say I am going to use a bad switch right. But, I am going to reduce or restrict my signal swing, this is not a particularly good idea as we saw in the analysis that if you reduce the signal swing the k t by c noise will increase.

If the k t by c noise increases, it means that you now have to use a larger capacitor to in order to maintain, the same signal to thermal noise ratio which means that the displacement current through the capacitor is now increased and you are pretty much back at square one you understand.

So, this is general principle if you try and operate with two small signal level you will find that you are limited by I mean your SNR is now determined, by thermal noise. And in order to maintain that SNR you have to increase or scale up the impedances or scale down the impedances such that, the k t by c component, becomes sufficiently small. And this will basically mean that you are paralleling many identical circuits together, which means that you have a larger power dissipation.

So, always you pay a penalty for working with small swings, so you have to try and ensure that your signal swings are as large as possible and the challenge there, then becomes that you need to design amplifiers or circuit blocks which are capable of handling large signals, without distorting too much.

(Refer Slide Time: 05:29)



So, the last couple of a points, I wish to make with regard to the flash converter architecture is the following. And if we got into this flash converter simply, because we were discussing, a delta sigma and the quantizer in the delta sigma loop, needed to be as fast as possible. And the fastest way of making a decision is the flash converter, so we went and got into the details and the nitty gritty of a flash.

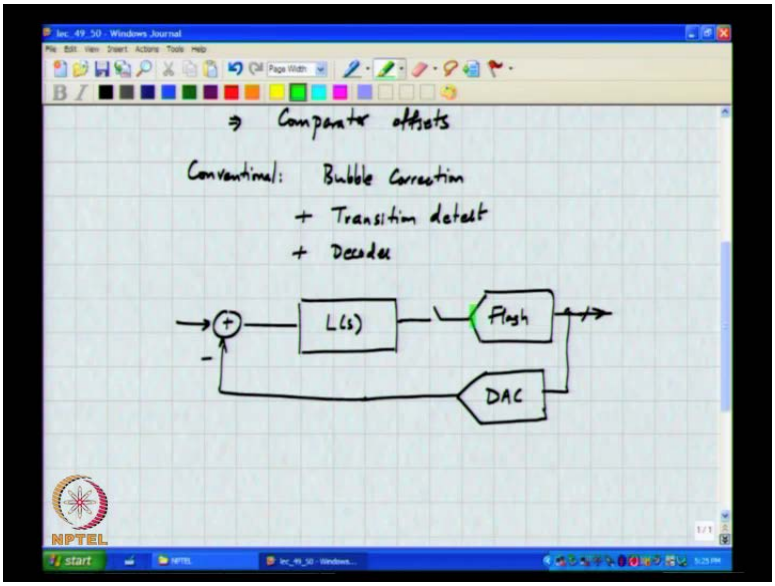
But, a flash converter as you can imagine is also used in other situations where latency, becomes a paramount issue and where you cannot afford, the delay or that characteristic of other A to D converter architectures. So, coming back taking a step backwards and then you know, after the discussion with on the flash architecture a couple of things that certain to flash converters, especially with regard to a delta sigma loop. One thing that we all know, now is that ADC non idealities, are noise shaped in a delta sigma loop right.

On the other hand d to a converter non idealities are definitely noise not noise shaped, because those errors appear as if they are added in along with the input. So, this means that you can get some what sloppy on the A to D converter design and there is also another aspect that usually the quantizer resolution in a delta sigma modulator is restricted to about 4 bits, there is no hardened fast rule or restriction, but 4 bits seems to be a good tradeoff between, complexity of implementation on one side.

And the benefits of using a multi bit quantizer on the other, so it is a good balance between the two, so you will find a lot of papers in the literature where the quantizer resolution is fixed at 4 bits.

So, this means that some things, which would not be practical in a higher resolution flash ADC, a standalone flash ADC typically would be a 6 to 8 bit converter. A 6 bit resolution is somehow very popular, for a whole bunch of flash converters less than that is too little resolution, more than that is within quotes to too difficult to do. So, in a lot of systems a 6 bit resolution is apparently enough in which case you go, for a 6 bit flash converter if latency is a problem or classic case in point is the destructive the channel system that I was describing to way earlier in this course.

(Refer Slide Time: 09:14)



So, one common aspect of these 4 bit flash converters used in delta sigma modulator is the following, so let us discuss comparative offsets, so if the comparator offset in a flash converter is large what do you think will happen.

Student: ((Refer Time: 09:47))

You might have the d n l problems and d n l, I n l problems.

You will obviously that threshold shifted from the ideal positions, so you will have d n l and I n l issues and if the threshold shifts too much, you will have a ((Refer Time: 10:05)) mean non monotonicity in the in the flash sometimes, you can even have a

missing code and so on. So, d n l and I n l problem is usually not an issue in a delta sigma loop simply, because non-linearity only causes the quantization noise flow to rise, I mean, so if you take a standalone quantizer the ideal spectral density is that is delta square by 12 divided by f s by 2, that is the noise spectral density of you know of the quantization noise.

Now, if the thresholds are moving around or have moved around from their ideal positions, then this noise floor will increase accompanied probably by also distortion harmonics and so on. However this increased noise is not really a concern, because it is all this is getting shaped out of the signal band. Now, the only thing that one needs to worry about typically is to make sure that the characteristic does not, become non monotonic and which can become a problem when the when the converter is embedded in a feedback loop.

So, what one normally does is you have the comparator array you have bubble correction plus transition detect plus decoder. When this flash embedded inside a loop for example, like this, now couple of things the sampling action itself where do you think one can realize the sampling action, I mean if the number of comparators what was the purpose of having a sample and hold in a flash converter.

Student: ((Refer Time: 13:32))

If skew comes

If the clocks is skewed, then you end up with poor dynamic performance and therefore, you have you know if you have a sample and hold that problem is avoided, in a delta sigma loop there are two things. One the flash is a very much smaller flash correct a 4 bit flash is four times smaller than a 6 bit flash for example and therefore, the speeds I mean, so the clock skews involved will be much smaller.

And to begin with you in a delta sigma loop you need to be worried about the entire loop as such, it is not just the flash alone. So just, because you can make a flash work at a very high speed does not mean that you will be able to put, that flash inside a delta sigma loop and make the whole delta sigma loop run at the same clock speed as a flash, which explains why you see.

If you look at the higher speeds of flash converters reported in the literature in a given technology, you will find that the higher speed delta sigma converters reported in the same technology are much slower, I mean if it was very straight forward to take the same flash and put it inside a sigma delta loop and make the whole loop run at a very high speed you know somebody or the other would have done it.

But, the problem with these very high speed flash converters is that the latency can be a significant fraction of the clock period, which means at the moment you try and embed this inside a delta sigma loop, you will have all sorts of problems associated with excess loop delay. So for, example if the clock if the latency of the flash is close to one full clock cycle, then the if you did not do anything of course, the loop would be unstable, you have to really compensate for a whole cycle of excess loop delay.

And of course, if the latency is greater than a clock cycle then you know at least what we have discussed in class does not work is in it. Because, that ensures that you need the data, before I mean how do you how would you solve the problem of excess delay in a continuous time delta sigma modulator you need a path direct path around the quantizer, where that direct path provides the first sample of the loop filter, so called loop filter impulse response, the loop filter is unable to provide anything simply, because it is excitation is coming, so late.

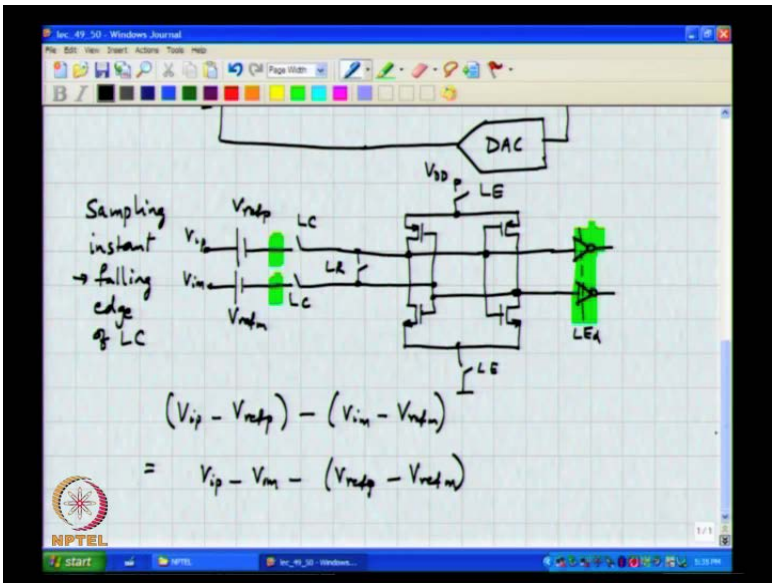
Now, if the delay or the latency of the flash itself is more than one clock cycle, then the direct part does not help either. Because, it is output is it the output is not being generated even after the end of one clock cycle you understand, so these kinds of problems make it difficult to clock a delta sigma loop, as fast as one would be able to clock a flash converter by itself.

So, flash ADC working in a inside a delta sigma loop will have first of all lot fewer clock skew problems, not only that the speed with which the converter is running itself will be slower, than would be you know possible for a pure flash where latency is not as big a concern. As it is for a delta sigma loop, this means that you do not need a you probably do not need a dedicated sample and hold, to sample the output of the loop filter and hold it, before going into the comparator array one can directly drive the comparator array using the implicit sampling in the latches itself.

The comparator array each lies you know can be concede of as having a pre amp and a latch, so and the latch is inherently sampling the output of the pre amp and making a decision. So, there is inherently sampling happening inside the latch, so one could say hey you know first of all I am running at much lower speed and clock skew is not an issue. So, I can simply my life by getting rid of the sample and hold, that is one simplification that is often made the next thing is to try and simplify the comparator itself.

The comparator is basically we have seen, it to be a pre amp and a Latch, the goal of the pre amp is to reduce the input referred offset of the Latch of the latch.

(Refer Slide Time: 18:50)



So, there are there have been realizations, where in an attempt to simplify the comparator one just does not bother with, a pre amp at all and simply uses the latch again I would like to bring to attention that several possibilities exist for the latch, it is not necessary. So, this is L C, Latch Reset, Latch Enable and when you hold this on some other delayed clock, so this way the sampling instant is the falling edge of L C.

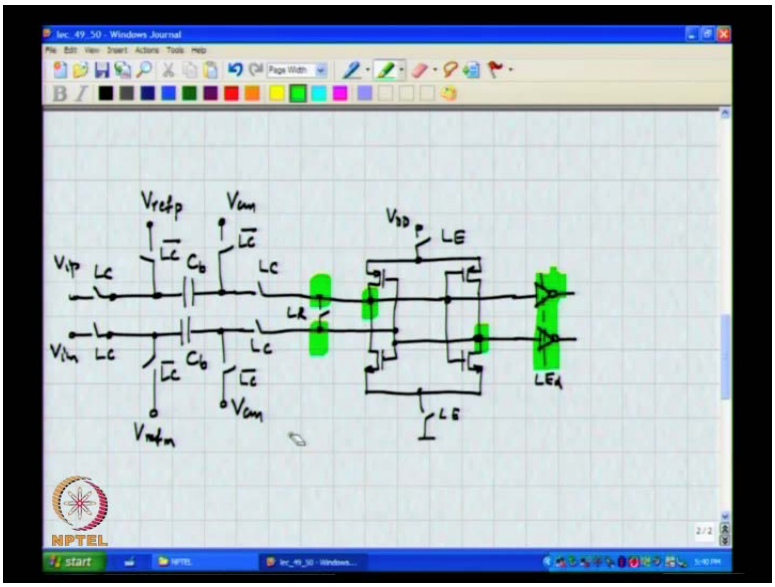
And the data is available at the rising edge of, I am sorry at the falling edge of L E D as we have discussed before.

Student: ((Refer Time: 21:06))

This is also a good point, this differential latch can only distinguish if these two inputs V in p and V in m, if one is greater than the other or not, one still needs to be able to subtract references. And one way of subtracting references is to say in principle what one needs to do is to apply, if one had a battery, this in principle would then detect this whether V I p minus V ref p is greater or V I m, sorry minus V ref m is greater and this is equivalent to comparing V I p minus V I m minus V ref p minus V ref m, does it make sense.

Now, can you suggest how one can subtract references from here, how can one implement these batteries.

(Refer Slide Time: 23:34)



So, the usual trick of having a capacitor, one important observation to make is that when are these battery is not needed or when are these capacitors needed.

Student: ((Refer Time: 24:12))

L C is off

You need these batteries only when L C is

Student: ((Refer Time: 24:14))

Off

Student: ((Refer Time: 24:16))

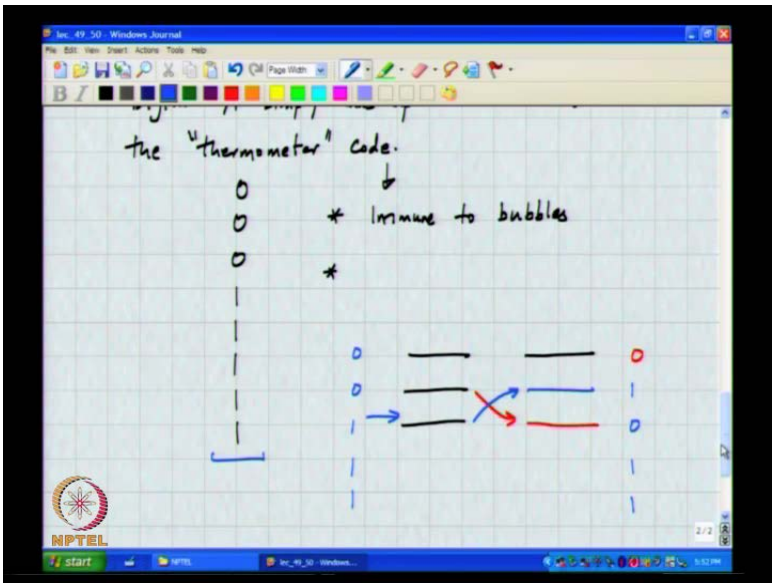
On sir

Is high correct when L C is not needed or when L C is low the latch is regenerating, so at that time these batteries are not needed which means that that is a good time to charges the capacitors which will invariably loose charge otherwise, because of all the switches connected to them. So, conceptually one to do this, so as this was for instance V I p, V I m, V ref p and V ref m, when L C bar is high what happens, when L C bar is high these capacitors are charged to, the capacitor on top is charged to V ref p minus V c m and the lower capacitor is charged to V ref m minus V c m.

When L C goes high, these capacitors are placed in series like this and the voltage here is if there is no parasitic capacitance at those nodes, then the voltage across the capacitor C let me call this C b, the voltage across the capacitor C b does not change at all. And the potential stored on the parasitic capacitance at these nodes will be V I p minus V I m minus V ref p minus V ref m, in assignment at given you another flavor of the Latch where again the input is subtracted, using coupling capacitors using a some somewhat similar switching arrangement.

Note; however, that we are still stuck with the offset due to the latch and if the offset becomes very large, then you can end up with very large errors in the thresholds. But, since the number of comparators is very few for example, a 4 bit flash requires only 15 comparators, one does not really have to implement bubble correction logic and you know the transition detect and decoder and such, one other way of generating the digital output sequence as a function of the thermometer code, that appears at the outputs of the comparator is to simply adds up the number of.

(Refer Slide Time: 29:02)



So, now if there is a bubble and if you add up the number of 1's appearing at the output of the comparator array, do you think it matters for example, ideally let us say this should have been the thermometer code. But, let us say for argument sake that this shows 0, if we normally took this thermometer code, without bubble correction logic to a transition detector and then, through the decoder you would get a completely erroneous value simply, because the transition detector which is accustomed to only seeing a single you know 1, 0 transition, now sees 2, 1, 0 transitions and therefore, is confused.

Now, what a if you simply add up the number of 1's in the thermometer code, what will you get.

Student: ((Refer Time: 30:55))

Now, it will just simply count the number of 1's in the thermometer code

Student: ((Refer Time: 31:05))

It is only 1 less than

You will only make a very small it will make 1, L s b error correct, so this way addition of the number of 1's in the thermometer code is gives you something, which is within codes in sensitive to bubbles without actually having any bubble correction logic. However, adding up in a regular flash where the resolution is not as low as 4 bits for

example a 6 bit flash converter, adding up all the 1's in a thermometer code would mean adding up 63 numbers.

Which means a whole bunch of adders and that sort of thing which is makes it not a very practical idea for large resolutions for smaller resolutions. However, if you have only 15 comparators adding finding the number of 1's in a thermometer code is a, is not a very expensive task from a view point of hardware.

How do you suggest you we add up the number of 1's

Student: ((Refer Time: 32:09))

Pardon

Student: ((Refer Time: 32:14))

I mean the easiest way of doing this is to add combine add up neighboring bits and then add up successive neighbors and so, on. So, you get a some kind of tree structure and you know with the depth of this tree is dependent on log n, where n is the number of numbers you wish to add I think this is called what is called a tree adder. So, this is something which is a also done in a flash converters, especially those used in a delta sigma loop.

Now, this addition of these 1's to give you the binary code, this addition does not have to be done within the loop you understand. So, the output of the sigma delta loop can be, simply the thermometer code that comes out of the comparator array as long as the DAC is happy to receive the thermometer code, the thermometer to binary logic can be done, outside the loop you understand and therefore, the latency involved in combining all these 1's etcetera is not a part of the loop at all and does not contribute to excess loop delay.

But, it is important that the binary, the DAC be able to accept a thermometer code it turns out that this is actually a blessing in disguise, because most DAC will be much happier with a thermometer code rather, than a binary code and so this works out quite nicely. So, the DAC is driven by the thermometer code and the thermometer to binary conversion is done, outside the loop does it make sense.

So, let me summarize this will make this immune to bubbles, can you comment on this solution and it is what this would do, if they offsets in the Latches were very large.

Student: ((Refer Time: 36:13))

It is more,

Pardon.

Student: ((Refer Time: 36:15))

towards.

Look carefully, let us say the offsets in the comparators were very large.

Student: ((Refer Time: 36:22))

If the offsets were large then you have problem in resolution of smaller inputs.

No, that is not what I am asking you I am asking you can you comment on I mean in a regular flash converter, where we have the thermometer code and the usual bubble correction logic etcetera. If there are big offsets in the latches offsets larger than an L s b for example, can you comment on what happens to the output code of the flash. Let us say you had the ideal comparator levels, thresholds were these and by some both of it this threshold remain like this and this threshold, became like this then what do you think will happen, if the input was here for example, what could what the omit code would we get ideally.

Student: ((Refer Time: 37:47))

1 1.

1 1 0 is what you would get, now what would you get.

Student: ((Refer Time: 37:53))

0.

Student: ((Refer Time: 37:54))

1 0 1.

You would get 0.

Student: ((Refer Time: 37:58))

0 0.

You will get 1, you will get 1, now what happens if the input happens to be here.

Actually I should have got 1 0 0.

Now you will get 1 1 0.

So, this basically I mean you can see that this causes, a fair amount of error and I mean this is a rather, let me take a even more pathological example, let us say this threshold moved here and this threshold moved there, then what will you get if the input was here, ideally you should have got 1 0 0 and now what will you get 0 1. So, if you did not have bubble correction logic you will find that, this gives you a whole lot of I mean pathological errors, I mean assuming everything below is, this is what you would get.

Now, if the comparator thresholds, now if you add up the number of 1's in the thermometer code the fact that the comparator thresholds move does not really.

Student: ((Refer Time: 39:53))

Actually it does not matter.

It does not matter.

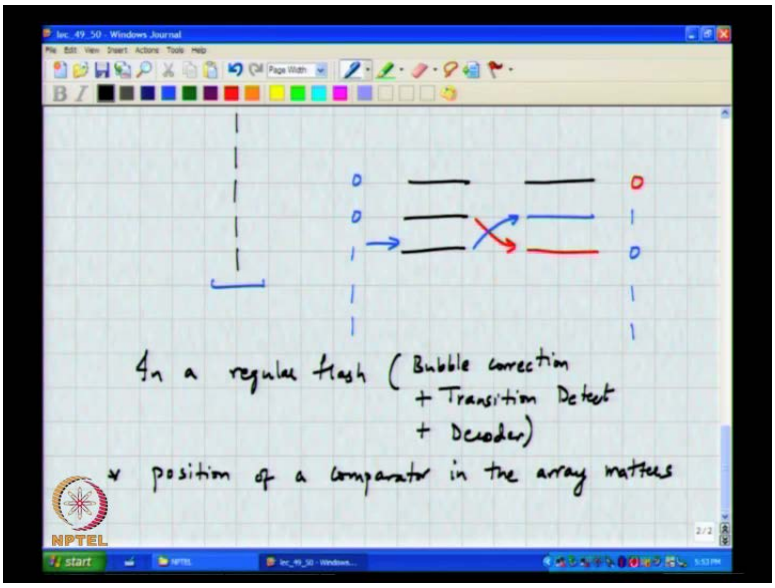
Is that is that clear.

Student: ((Refer Time: 39:57))

There are total number of n constants.

You understand see in a regular flash array the location of the threshold does matter

(Refer Slide Time: 40:19)

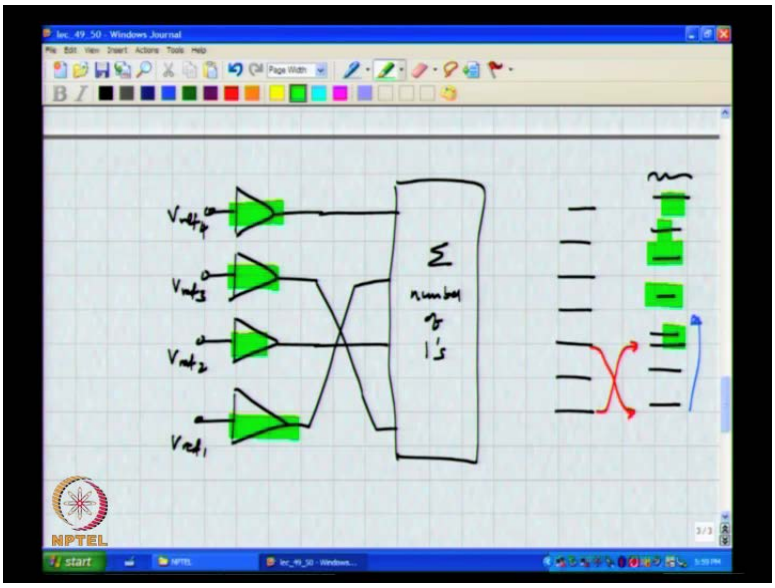


Because, the position of a comparator in the array matters in the sense that, if I physically move a comparator down, I have the array if I interchange the position of two comparators, while having the same references to the giving the same references to them what will happen to the output code. It will be in grievous error, because the logic following it does not know that the positions of the comparators have changed.

However, if you add the number of 1's in the thermometer code, you can move the comparators around without effecting the output code. Because, as far as the final output of the A to D converter is concerned, every comparator output contributes in the same way, because it is symmetric from the output of the path from the output of any comparator to the final output is logically, the same you understand.

Because, all that is happening is that if the output of a particular comparator is 1, it contributes 1 to the final code. But, that is not the case in a conventional flash array where you do this transition detect from the all 1's to the all 0's there the outputs of several comparators, I mean the 1's which are near the transition point or the 1's which really matter and therefore, the position with which I mean the position of a particular comparator does effect the output code is this clear, If I take I mean if I in other words.

(Refer Slide Time: 43:04)



So, let us say all these and then this goes to the digital backend conventionally, if I now swap this two wires for example, what do you think will happen. The output code will be in error is in it whereas, instead of having the conventional digital backend, if I just simply added the number of 1's in the thermometer what would happen, there will be no change in the output code.

Now, changing the position of these comparators can be thought of as change in offset you understand, so in other words you were trying to realize comparators with these thresholds, due to offsets I got this let us for argument sake say this comparator threshold became this and this threshold, became this and this moved and this moved and so on. Now, you do not know which threshold came from where, it could be that this threshold actually was this comparator is threshold which is moved, so much lower by due to offset.

But, if you add the number of 1's in the thermometer code what happens, it does not matter, it appears as if you had a comparator array with these thresholds correct is this clear. So, regardless of the amount of offset in these comparators the characteristic will always be monotonic correct you understand. So, if the input increases you cannot have a situation where the output code, comes down having a monotonic output is important, because this is part of a feedback loop, otherwise the feedback loop basically you will get confused.

It is attempting to increase the output and then it finds that in spite of going in the same direction, the output actually decreases and then it does not know whether it has to go in the same direction, I mean how does negative feedback work you compare the output with the input and tweak in a direction, so as to reduce the error. If this the loop quantizer is non monotonic, then you can expect trouble in the sense that you increase and then the output actually decreases, contrary to your intuition, so then that makes the loop confused you understand.

So, adding up the number of 1's in a thermometer code is a good way of eliminating, non monotonicity irrespective of the amount of offset of course, there you will still be stuck with d n l, I n l problems correct. Because, these thresholds spacing are not uniform, however, the requirements on offset can be greatly relaxed to get a certain I mean to get non monotonic behavior, earlier you know small amount of offset was enough to cause this kind of problem.

Now, even though you get d n l, I n l you can rest ensure that even if you have large amounts of offset they will not be any non monotonicity in the quantizer does it make sense. So, these comments kind of summarize, whatever I had to say on flash converters as well as those aspects of a flash converter, which are relevant to a delta sigma modulator loop.

And then in the next class we will start with the other part of the puzzle which is D to A conversion. So, we have made one half of the quantizer, which is the A to D converter we need to make the you need to understand what has to go into the D to A converter, which takes the output code of the flash and generates a current or a voltage, which is you know then fed back into the loop filter.