Circuits for Analog System Design

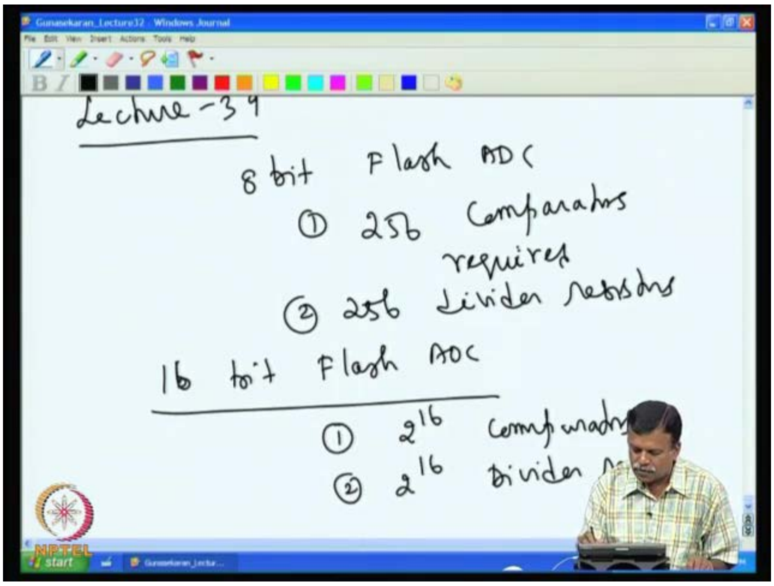
Prof. Gunashekaran. M. K.
Center for Electronics Design and Technology
Indian Institute of Science, Bangalore

Lecture No. # 39

Flash ADC and ADC Converter Errors

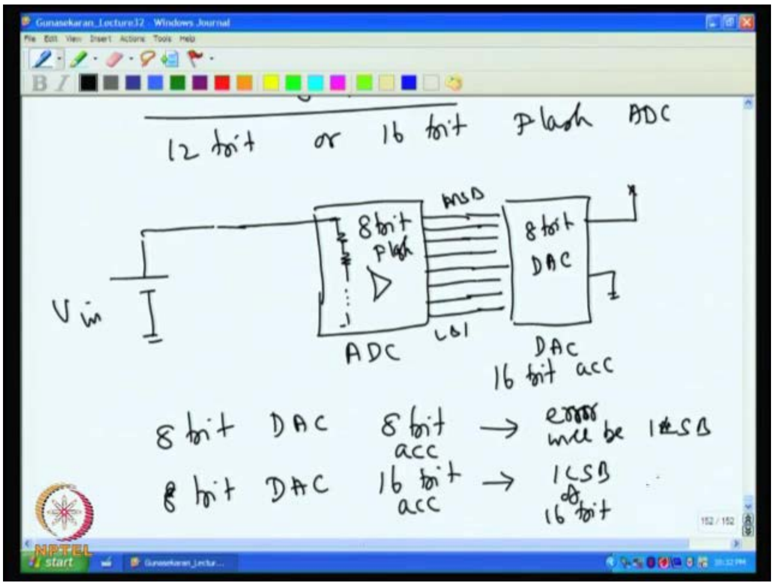
In this lecture, we continue to talk more about flash A to D converters, because in the previous lecture we had discussed about 8-bit flash A to D converter.

(Refer Slide Time: 00:32)



And, we had shown that for 8-bit flash ADC, 256 comparators required. And then, similarly, you need 256 divider resistors. These two are very tough to achieve, because if you can have 256 comparators and 250 divider resistors, it is somewhat manageable. But, then if you want go for 16-bit flash ADC converter – you look for this, then I need 6400 – 2 power 16 comparators. And then, similarly, 2 power 16 divider resistors. This is not easy to achieve, because 64000 comparators and then 64000 resistors; and then, combining the 64000 bits into 16-bit usable A to D output, it is not an easy job. So, normally, 16-bit flash ADCs are normally not made this way. So, for this, only sub-ranging ADC is used.

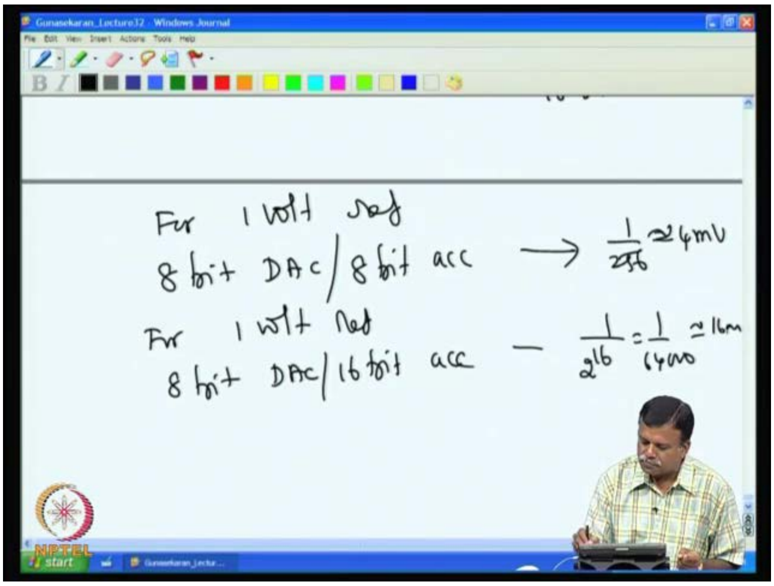
(Refer Slide Time: 01:59)



Sub-ranging ADC is used for 12-bit or 16-bit flash ADCs. Let us we what is the sub-ranging flash A to D converter. Sub-ranging flash ADC converter – what is done is that you have the input voltages to be digitized – V input; that is actually given to A to D converter, which is an 8-bit flash. So, you have an 8-bit flash here. Basically, it goes to a resistor divider here, so on; and then, you have comparator. So, basically, it is an 8-bit flash ADC. This gives an output, which is 8 bits. So, you have an 8-bit output here.

You have an MSB here and then LSB here. This actually is given to again 8-bit DAC (D to A converter), which actually gives you output voltage corresponding to 8 bit. So, what you have done is you have actually put A to D here; then, you have put a D to A; then, again you got back the analog voltage. But, then this is only 8-bit output. But, this DAC – even though it is an 8-bit converter, 16-bit accuracy is maintained in the sense that if I have an 8 bit… For example, if I say 8-bit DAC, then 8-bit accuracy. If I took 8-bit accuracy, then the error in this for example, if it is 1 LSB, error will be 1 LSB maximum; whereas if I look at 8-bit DAC, then if it is 16-bit accuracy, this will have an error, which will be 1 LSB of 16-bit. This is (Refer Slide Time: 04:45) 1 LSB of 8 bit.

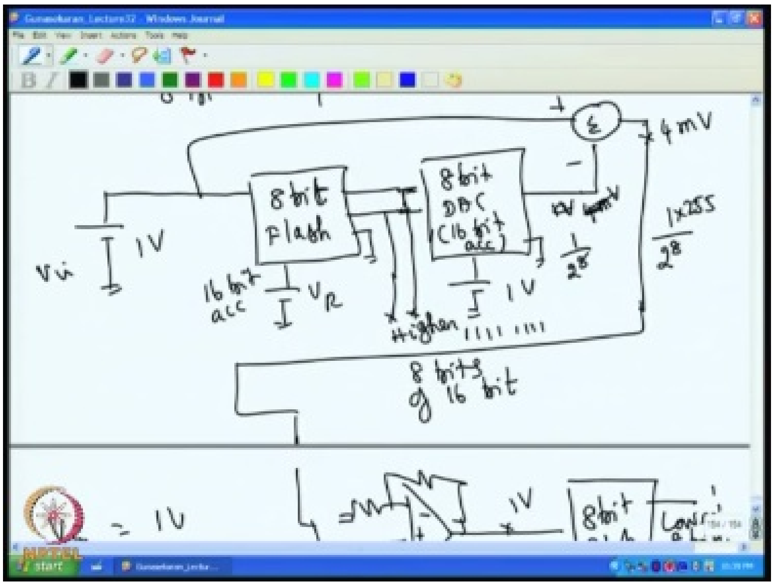
(Refer Slide Time: 05:01)



For example, if I take 1 volt as my reference, for 1 volt reference, 8-bit DAC / 8-bit accuracy, the expected error is 1 by 256; that is, roughly equal to 4 milli volt. For same 1 volt reference, 8-bit DAC / 16 bit accuracy; error will be 1 in 2 power 16; that is, 1 in 64000; that is roughly 16 micro volt. So, that is the meaning of 8-bit DAC with 16-bit accuracy. So, we have a DAC (Refer Slide Time: 05:50) here, which is more accurate than the normal 8-bit DAC; that is, it is 8-bit DAC, but it is accurate to 16 bit.

Now, what we can do is this analog output, which is (Refer Slide Time: 06:04) very accurate can be now subtracted from the input and then we can make another 8-bit converter. So, I will show you the figure in the following page.

(Refer Slide Time: 06:19)



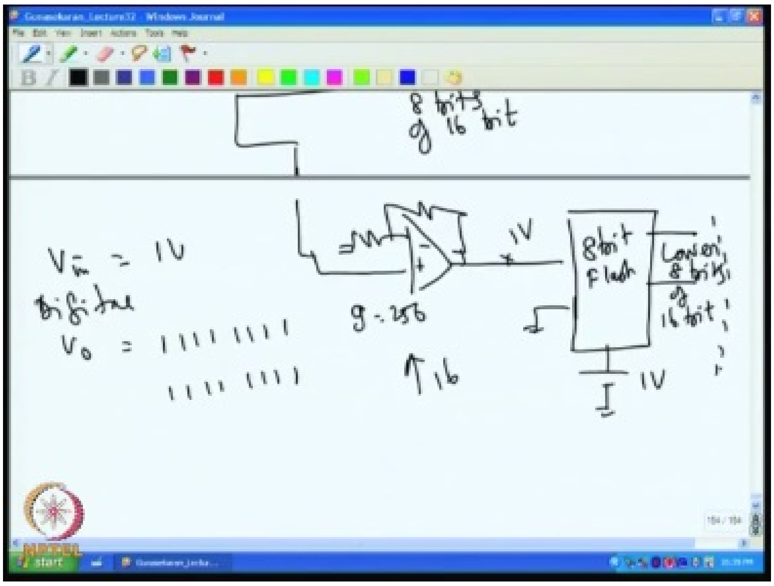
What you do is that you have the input voltage V input. Essentially, you give it to… I have a V reference here. And then, I give it to input. So, this is an 8-bit flash. And, this actually gives an output to 8-bit DAC with 16-bit accuracy. This gives me the… This has a V reference and then this gives me output. This output – what I do is I will give it to a subtractor amplifier here. The difference between the input and this is appearing here. And, this – what I do, I will actually now amplify this voltage. So, I can amplify the difference voltage. So, we can get into the next page.

We will have the amplified voltage coming here (Refer Slide Time: 07:40). Here we amplify this. Say for example, if another 8-bit converter I need; amplified by 256 times gains equal to 256. Then, whatever the output is coming, this I give it to 8-bit flash. And, this 8-bit will act as a lower 8 bits of 16-bit converter. This will act as a higher 8 bits of 16-bit converter. So, you have the lower 8 bits here and then higher 8 bits here. So, you have the 16-bit conversion. But, then we have used only here 256 resistors and the 256 comparators; and then, here again, we used 256 comparators and 256 resistors. So, it is only totally 512 comparators and 512 resistors; not 64000 resistors and comparators.

Now, how this is working? If I give for example, 1 volt as input, did my reference all taken as 1 volt, even for this also I take as 1 volt, then if you analyze the circuit… That if you have 1 volt, then the 8-bit converter would have converted this; and, that given all of them equal to 1 1,1 and 1 1 1 (Refer Slide Time: 09:43); and, of course, this also has to be 16-bit accuracy; 16-bit accuracy is required. Then, we should have given all of them 1 1 1. But, then the final output when this is converted, this is not going to be 1 volt; this will be actually 1 volt divided by 2 power 8 as 1 LSB. So, 1 LSB error would have been left and would have got 4 milli volt output from…

Here (Refer Slide Time: 10:10) you would have got 1 volt divided by 2 power 8, is 1 LSB. And, totally, 255 LSB is there. So, if we take this, the difference will come here as 4 milli volt. There is 1 volt and even all bits are high, they are not going to get 1 volt; that will be 1 in 2 power 8 into 255; that will give error of 4 milli volt. This 4 milli volt is actually amplified by 256 and that actually will give you 1 volt. Now, this 1 volt will be digitized by this converter; and then, you will all get 1 1 1 1. All 8 bits will be 1. So, essentially, given V input as a 1 volt and then the digital output digital output coming as… All the lower ones are 8; and, the upper ones are also coming as 8 bits; all high. So, that way, with less number of comparators and resistors, one can get the required 16-bit converter.

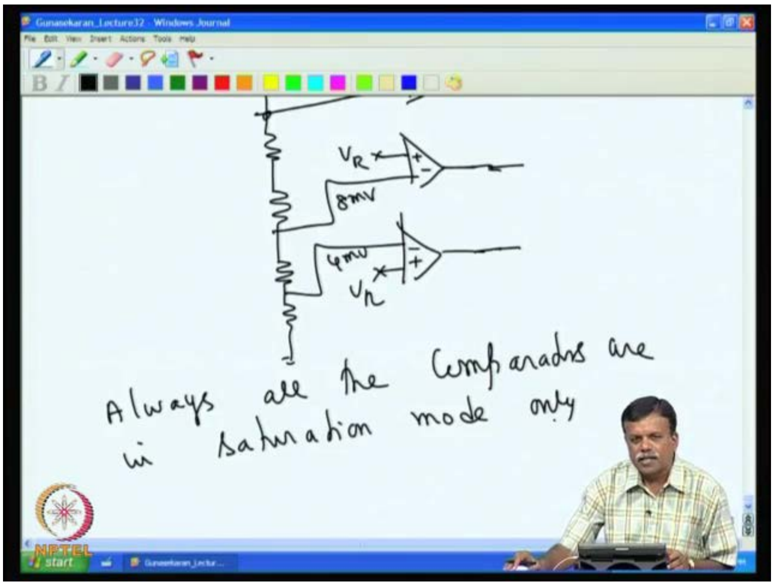
(Refer Slide Time: 11:33)



Of course, the conversion to be done in two stages and amplifier to be made… For example, if I want 16-bit accuracy, then I can have a gain of 16 here. But, 12-bit accuracy – suppose if I want 12-bit accuracy, then I can have a gain of 16 here. And then, I can put the 4-bit converter here. Again, the 4-bit will give me 1 1 1. So, this is actually called sub-ranging A to D converter, which is very common in flash ADC, because there is no way one can get 64000 comparators and 64000 resistors and make them workable units. It is practically not viable. So, almost all higher bit converters, flash converters are all actually the sub-ranging A to D converters. Sometimes it is also called as pipelined ADC.

These A to D converters are very fast, because you have only the direct comparisons with comparator if the conversion time goes in nano seconds. So, the advantage is (Refer Slide Time: 12:45) very short conversion time. The flash ADC converter – short conversion time, but it is costly.

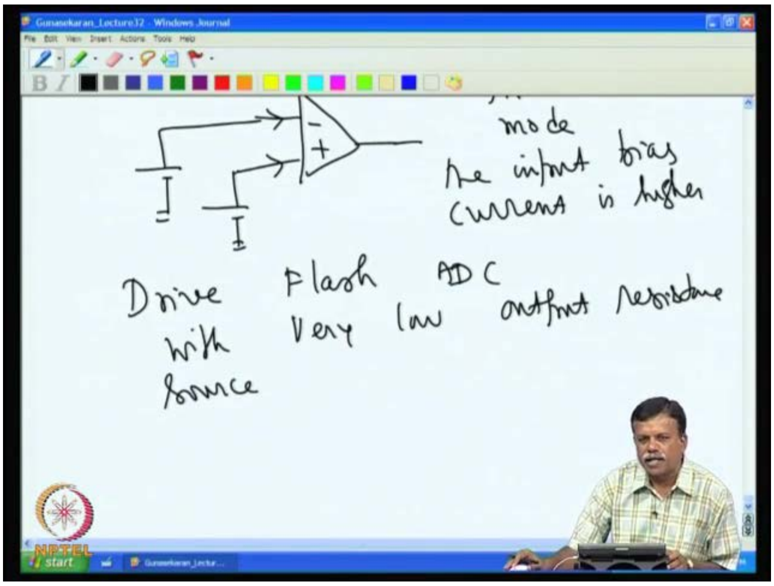
(Refer Slide Time: 13:15)



Then, other advantage that we will see in the converter is that we have to have the inputs – are compared with so many comparators, because the other major disadvantage is that we have for example, the input voltage is divided by so many resistors; and then, we are also putting so many comparators comparing with the reference. So, we have this; we have a reference; we have a V input here; and then, we give this one to the second one; and then, we have V reference here, V reference here. So, we have so many comparators and so many reference voltages. This is again given to V reference. Say if you look at the output, you have plus minus; you have plus minus; and then, the input will be minus and plus.

Now, we have the output voltages. This is V R (Refer Slide Time: 14:25). Now, if the input voltage… For example, if you take the comparators; for example, if I have 1 volt reference and if this is 4 milli volt and this is 8 milli volt and so on, you see almost all comparators are in the saturation stage only. So, half the comparators is suppose only the 0.5 volt, half the comparator will be high and other half will be low. So, all that time, the comparators are all in the saturation level only. Always all the comparators are in saturation mode only; that means the input impedance seen by the comparator is not known, because…

(Refer Slide Time: 15:30)

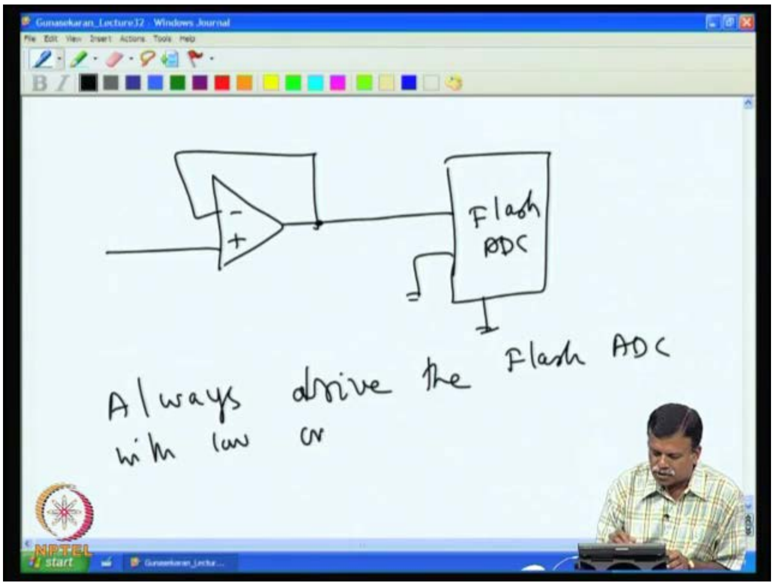


Our assumption that the input impedance is very high for the op-amp is valid only if it is in the linear range, because if we take any comparators, we have a plus, minus; and then, we are comparing two voltages; then, finishing the linear region, the bias current is what is a taken at the input. But, if it is in the saturation region, this bias current is higher. So, in saturation mode the input bias current is higher. So, essentially, what happens, when the input voltage changes, the current taken by each comparators changes? So, that actually is the problem, because we have to make sure that this current is very small. The current actually taken by the op-amp is very small. This is to be ensured.

This is one of the major problems because when input changes, the current taken by the comparator changes. So, it is essential that we drive the flash ADC with very low output resistance. So, it is essential that to drive the flash ADC with very low output resistance source; otherwise, the varying current, that is, the input impedance offered by the flash

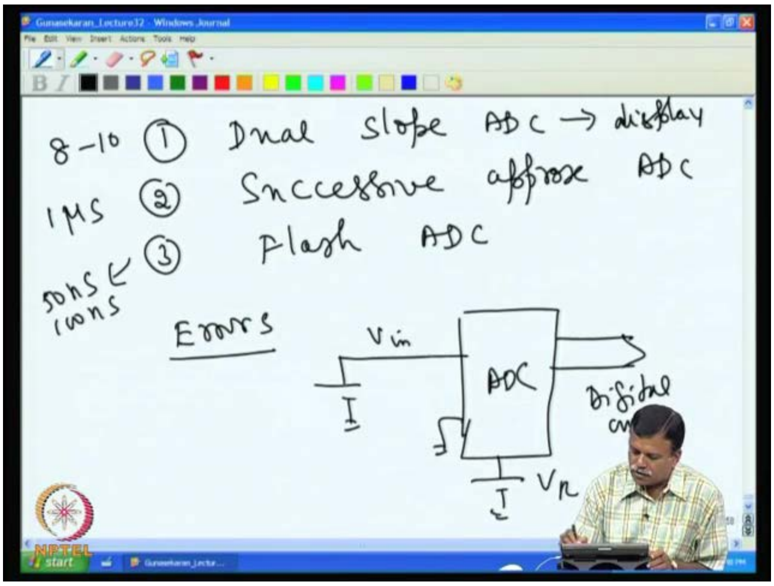
ADC continuously varies with respect to input voltage itself. So, it is essential that we drive this kind of flash ADC with very low output resistance.

(Refer Slide Time: 17:25)



That means if I have a flash ADC, then as an input – I give this one; that is, in the sense we have a voltage follower at the input. So, the output (( )) by this is very low. So, always drive the flash ADC with low output resistance. This is one of the major issues. And, most of the converters now have inbuilt voltage follower, so that this problem is taken care. So, this is flash ADC. And, the flash ADC you have even today – even the nano second converters – 100 nano second conversion time, even 50 nano second conversion time, flash ADCs are available. Compared to dual slope and successive approximation ADCs, they are much costlier. So, if you look at these A to D converters that…

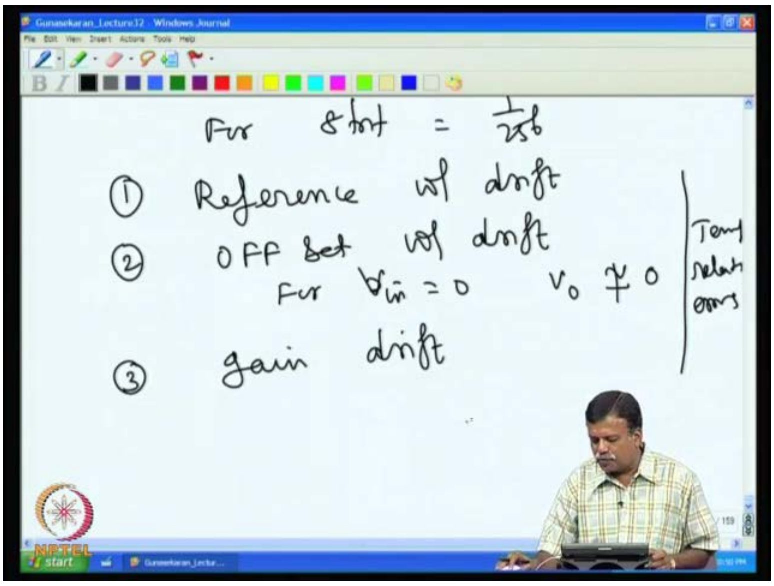
(Refer Slide Time: 18:49)



We have seen three A to D converters so for; that is, one is the dual slope A to D converter – dual slope ADC; and the second one is successive approximation ADC; and then, third one is flash ADC converter. Now, this is a used mainly for displays, where the conversion is very slow. So, you have 8 to 10 conversions per second; that is all possible. This actually have 1 micro second level of conversion time; that is, successive approximation A to D is much faster. It is about one micro second conversion time. And, the flash ADC – it will be about 50 nano second to 100 nano second conversion time is possible.

Now, in all these things, if you see that we have issues, because after all this A to D converters if you see, they have a voltage dividers, are comparing the voltage and so on. The final conversion is achieved. They all have errors. So, if you look at the errors involved in all these A to D converters, that basically you have a problem due to… For example, if I have A to D, we have a (Refer Slide Time: 20:27) reference voltage, and then we have an input voltage, and then digitized output – digital output is coming here.

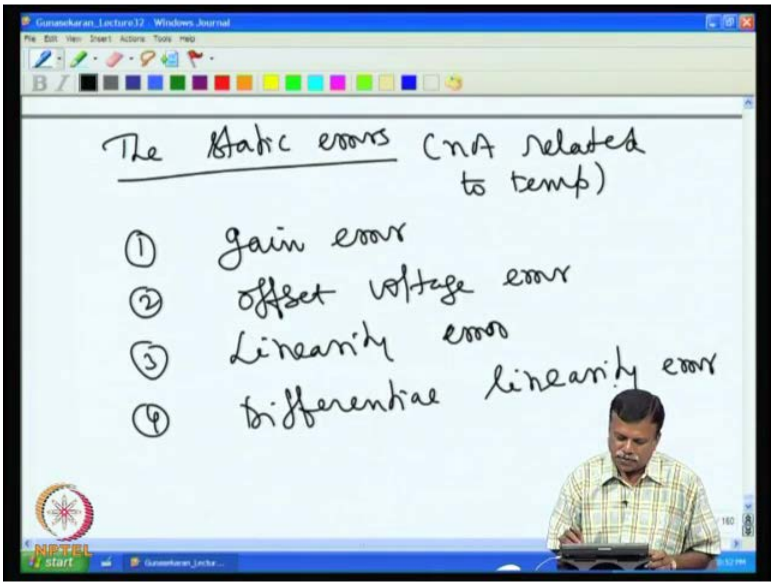
(Refer Slide Time: 20:49)



And then, one thing is the resolution. Resolution is actually resolution of this A to D converters. Resolution is actually 1 in 2 power n. For example, for 8 bit, it is 1 in 256. So, resolution actually you will get depending upon the number of bits. If 8 bits, means you have 1 in 256; if it is 16 bits, it will be 1 in 64000, is the resolution. But, it does not mean that you will get the accuracy also that much. One main problem with the A to D converter error is that you have a reference voltage drift. We are using a reference voltage; and, the reference voltage is the main input based on which only all conversions are done. So, if you take reference voltage, that (( )) creates error in the output. So, the errors will be: one is reference voltage drift – this directly affects the output; then, second one is offset voltage drift – because if we have an offset voltage, the zero input, zero output is not the case in most of the converters. For example, if we take the converter for 0 volt, V in is equal to 0, V output need not be 0. This is actually an offset voltage. But, this offset voltage changes with temperature. So, this is another major error one has to worry about this.

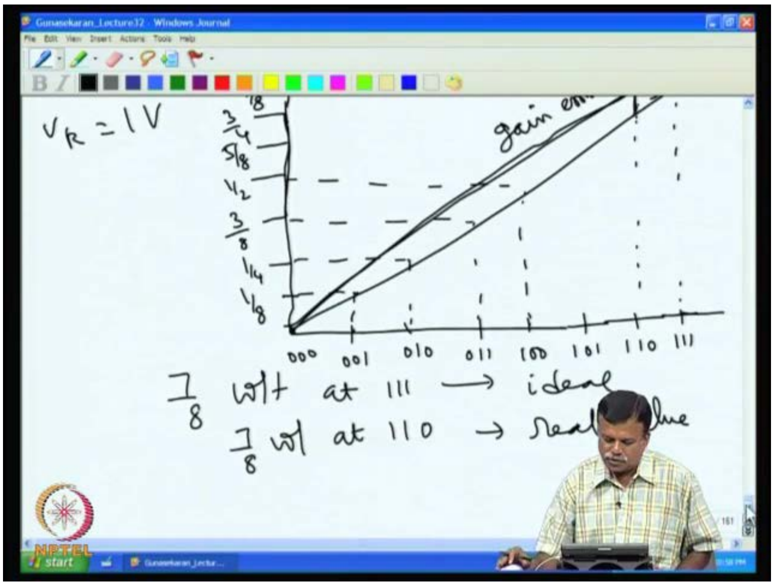
Then, the third error is gain drift. The gain of the ADC also changes with temperature. So, these are all actually temperature-related errors (Refer Slide Time: 22:51). That is, when temperature changes, this reference voltage drifts and that produce error in the output, offset voltage drifts, and then the gain also drifts. And, that produces temperature-related error.

(Refer Slide Time: 23:30)



Now, without temperature-related error, if you look at what is the inherent A to D or D to A converter errors, they are all actually like this. So, the static errors – that is not related to the temperature. That errors are: one – you have gain error, offset voltage error; then, linearity error; then, we have differential linearity error. They are always is nothing to do with temperature. So, at any given temperature, you have these errors. And, these errors of course change with temperature; of course, we are not much worried about differential linearity error changing and the linearity error changing with temperature. These two errors actually gain error and offset voltage error changing with temperature is an issue. So, what are these errors and why one has to worry about this?

(Refer Slide Time: 25:12)



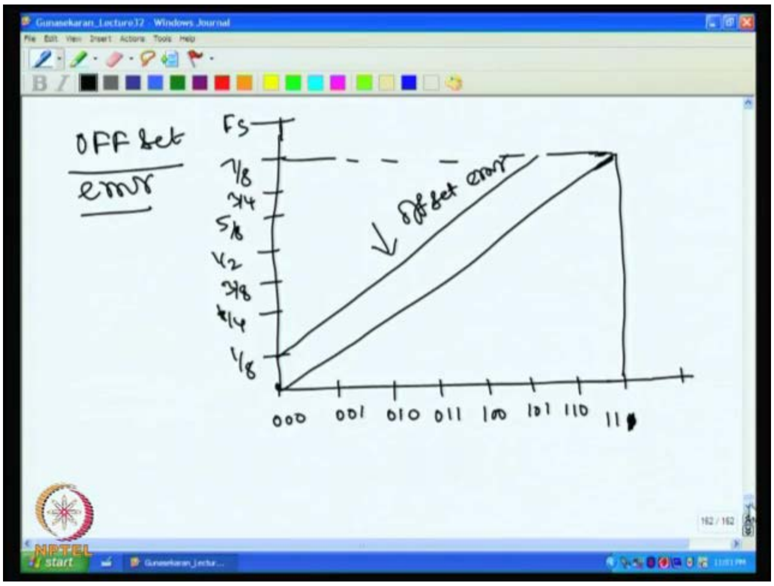
Now, if you look at the gain error, the gain error looks like this. For example, if I have an ideal converter, we take to explain this 8-bit converter and then explain these errors. Take for example… Instead of 8-bit converter, take 3-bit converter. For example, if I take a 3-bit converter, which has 1 1 1 as a digital output, if I have to start from 0 0 0, then the next code will be 0 0 1. Then, I have 0 1 0, then 0 1 1, then 1 0 0; you have 1 0 1; then, we have 1 1 0. Final one will be 1 1 1; that is the last one. Now, corresponding output for example, if I have an output, that 1 bit will have 1 by 8; then, you have 2 by 8, that is, 1 by 4; then, I will have 3 by 8; then, half; then, we have 5 by 8; then, 6 by 8, that is, 3 by 4; then, we have an example with 3 bit. Then, we have 7 by 8; then, full scale. So, if I plot the curve, that ideal case if this is full scale, we will have full scale voltage.

For 7 by 8 actually if we take, this is the full scale voltage. It is here (Refer Slide Time: 27:14). But, you will have for full scale, it is at this point and it goes meets. This actually will be meeting 7 by eighth point. This is the ideal output. For example, if it is 0 0 0, you are supposed to get 0 volt (( )) 0 1 1 is supposed here. 1 by fourth if you compare this; and then, 3 by 8 would have been this. For half volt, it should have been… Here V reference is taken as 1 volt. Now, this is the ideal converter that you get. When 1 1 1 is there, we are supposed to get 7 by eighth of a voltage. If it is 1 voltage – full scale voltage – that 1 1 1, we are supposed to get 7 by 8. And finally, it reaches 1 volt. 1 LSB is left out to get the full scale voltage of 1 volt.

Now, in actual case, if the gain error is there, instead of meeting it here (Refer Slide Time: 28:25) for example, 1 LSB gain error if it has, then this gain will be meeting here. So, you will have the gain error coming into this. Now, this is a gain error. So, you get a gain error, which is not, because instead of meeting 1 1 1, instead of reaching… 1 1 1 is supposed to meet here, but it had met at this point only. So, you have a gain error, because probably you will get a full scale voltage, which is 1 volt at 1 1 1 itself instead of 7 by 8. So, if you look at this relation, then 7 by 8 volt at 1 1 1 is what suppose to get. But, if it is 1 LSB gain error, then you will have 7 by 8 volt at 1 1 0 itself reached. It is supposes to be ideal case. This is what you got it. So, you got a 1 LSB gain error at the output, that is, at the maximum value, you have 1 LSB gain error. So, this is one type of error.

This is coming, because the output voltage what we are getting, the multiplication factor is not correctly set. That is the reason to get the gain error. This is correctable actually. Many of them give provision for correcting this, but many converters – there is no provision to correct any of these errors. So, one have to be aware of this possible gain errors. Of course, ones solution to correct this would be to adjust the V reference, so that the gain error actually goes off.

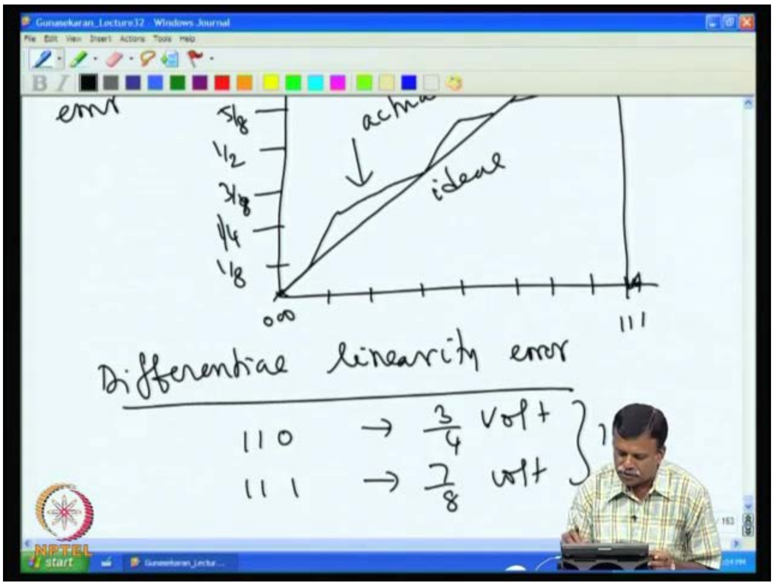
(Refer Slide Time: 30:55)



Now, the other error is similar way if we can extend, for example, offset voltage error, we can have the same thing. Take the offset error that we have – similar (( )) 0 0 0 if you take and so on, we can have 8 bits. So, we have the last one – for example, you have 1 1 0, and then the last one 1 1 1. So, you have 0 1 1, 1 0 0, 1 0 1. I have I think 1 1 0; I think 1; this is correct here. So, we have here 1 1 1. So, we have… Now, corresponding to that, we will have the same output. So, 1 by 8, 1 by 4 and so on; we have finally 7 by 8 and then full scale. So, 1 by 4; then, 3 by 8, half; then, you have 5 by 8, then 3 by 4, and so on; we have this. So, if you have the… we have 1 0, 1 1.

Here there is the missing point; I will change this. Here 1 0 1 (Refer Slide Time: 32:50). This is correct. Now, we plot the… Regular expected output would be like this. So, we will have a full scale voltage. So, 7 by 8 actually supposed to meet here; and, this supposed to meet 1 1 1 here. Because of offset voltage, it could be going like this. Here 0 0 – it is not actually 1 by eighth. So, this is offset error. Offset error is correctable; one can actually… If the provision is given, one can adjust the offset and make it 0. But, the offset drift is not correctable; there is no provision to adjust also. In many converters, even an offset voltage adjustment is not possible. This is offset voltage error.

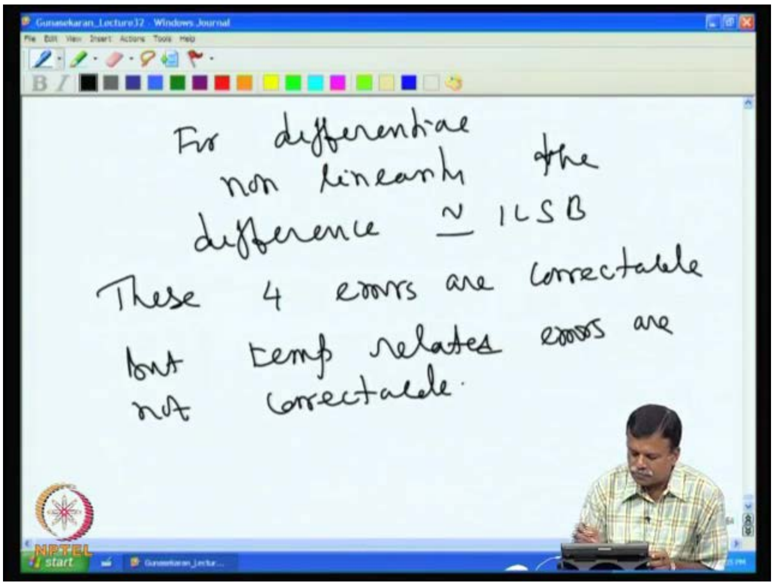
(Refer Slide Time: 34:00)



Then, we also have linearity error. Now, what is done is the same converter if I plot, you will have 0 0 0 as an origin. I will put the… Assume this is 1 1 1; then, we have here 1 by 8, 1 by 4; then, you have 3 by 8, then half, 5 by 8, 3 by 4. Then, we have 7 by 8 and then full scale. Here we are supposed to go like this and then we are supposed to get 1 1 1 at this point. This is linearity error. This is an ideal curve. In actual case, you may have like this coming back, then going, coming back; even it can go minus and come back. This is actual. This is actually linearity error. Similarly, you will have problem with differential linearity; that is, adjust ones are not going by 1 LSB; that is, two outputs; or, may not be showing exactly 1 LSB change.

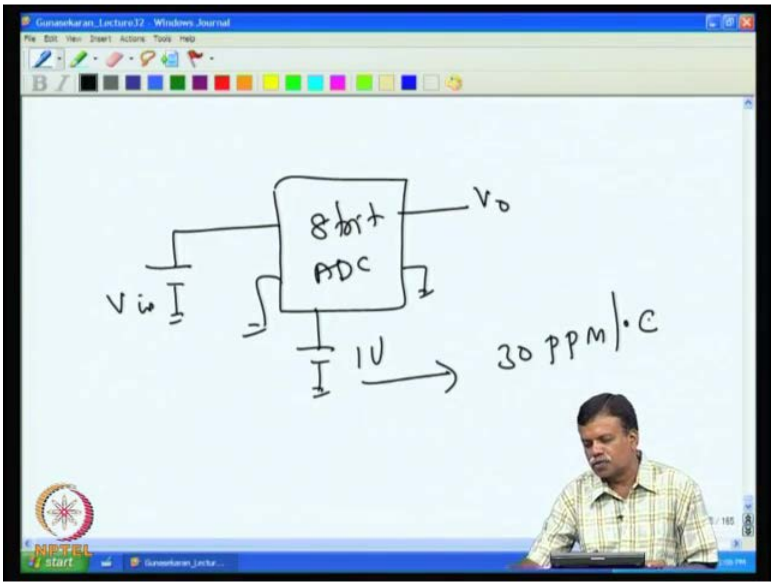
Differential linearity error (Refer Slide Time: 35:54) – for example, if I have say 1 1 0, 1 1 0 is supposed to give you three-fourth of a volt; and, 1 1 1 – supposed to give you 7 by eighth of a volt; the difference is 1 LSB. But, the difference need not be 1 LSB. In ideal case, this can be…

(Refer Slide Time: 36:34)



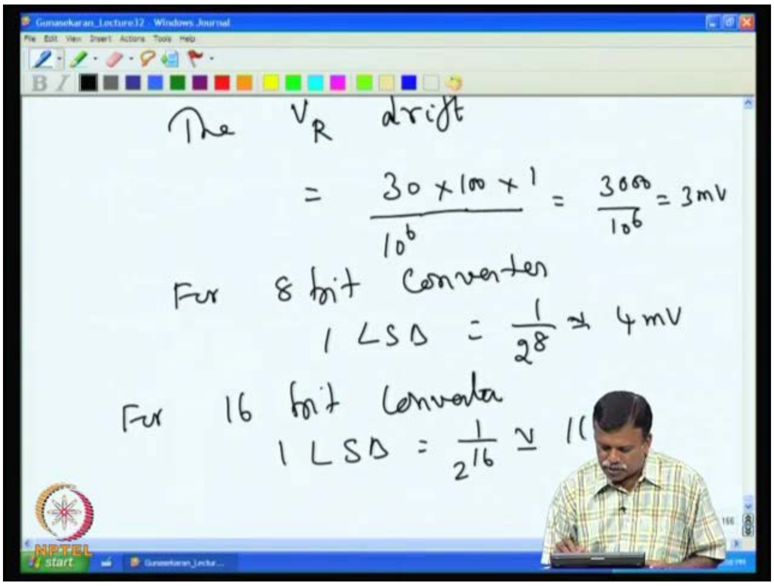
For differential non-linearity, the difference will not be equal to 1 LSB. This is actually coming as a differential linearity error. So, these errors are actually fixed. And, these four errors, that is, difference linearity error, linearity, and then offset voltage, and the gain error; they are not serious issue, but they can be corrected. In most of the converters, it is possible. But, the temperature-related error is a serious issue. So, these four errors are correctable. But, temperature-related errors are not correctable. So, one has to worry about this; for example, if I have A to D converter, which has 1 volt reference, then 1 volt would be the analog expected output voltage.

(Refer Slide Time: 38:20)



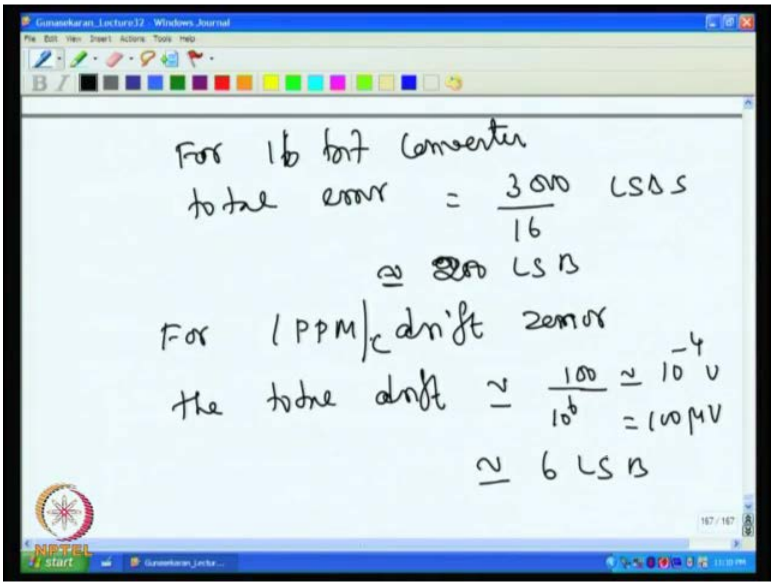
Now, let us see how these errors come into picture in real converter. For example, we take A to D converter, which has 1 volt reference. And then, I have V input here; and then, you have a V output. So, we take 8-bit ADC. If we take regular zener, we have for example, 30 ppm per degree c drift zener is used.

(Refer Slide Time: 38:46)



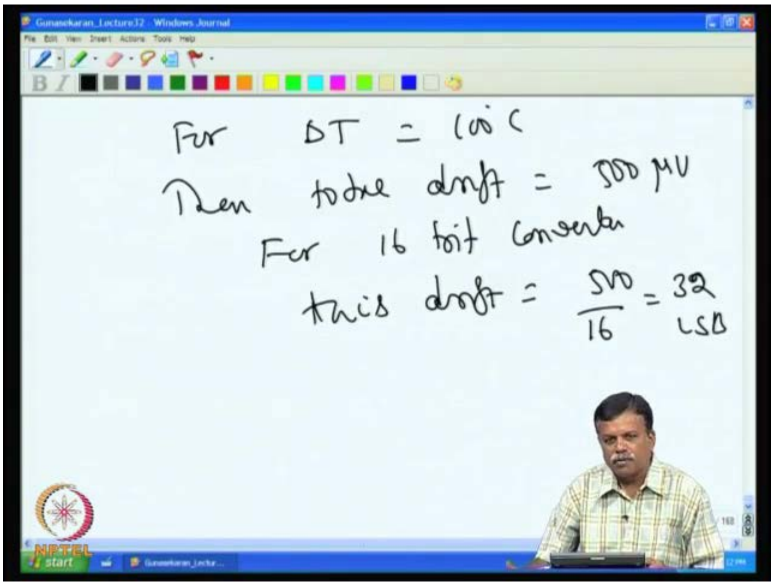
Then, we will have for 100 degree c; for delta T is equal to 100 degree c, the reference voltage drift would be V reference drift, would be 30 ppm into 100 degree c. So, we put 100 degree drift. And then, if it is 1 volt, then we take this 1 volt. So, that would be the actual drift; that will be 3000 divided by 10 power 6; that is equal to 3 milli volt. So, if it is an 8-bit converter, for 8-bit converter, 1 LSB would be 1 volt divided by 2 power 8, which will be nearly equal to 4 milli volt. So, this 1 LSB error is… This V reference drift is coming less than 1 LSB. But, for 16-bit converter, then 1 LSB is equal to 1 in 2 power 16. That comes to be about 16 micro volt. Then, this 3 milli volt drift will be very huge; that means if you want to go for 16-bit converter, then the reference also has to be stable; then only 1 LSB error is possible.

(Refer Slide Time: 40:47)



Then, for 16-bit converter, total error would be 3000 divided by 16 LSBs; that would be equal to nearly 200 LSB error. You see that soon the… When you go for higher and higher bit converters, even the reference need to be very stable; otherwise, there is no point in claiming 16-bit accuracy if you are using this 30 ppm reference voltage source. So, one have to go for 1 ppm reference source. For 1 ppm per degree c drift zener, the total drift would be equal to… You will have 1 in 10 power 6 into 100, is the voltage drift. So, that would be equal to 10 power minus 4 volt; that is, 100 micro volt; that is even equal to 6 LSB for 16-bit converter. Even for 1 ppm reference, we can get only 6 LSB. So, getting 1 LSB accuracy in 16-bit converter is actually very difficult. So, one has to be aware of this problem due to the reference voltage drift.

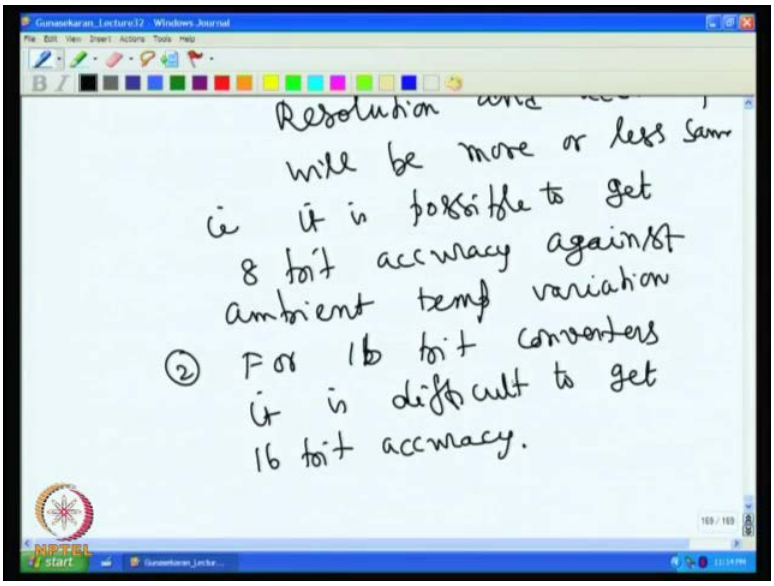
(Refer Slide Time: 42:54)



Similarly, offset voltage drift is another major error, which actually affects the converter's accuracy very much. So, the second voltage is offset drift. Offset drift is given in terms of micro volt at the input. So, most of them give offset voltage, is given in terms of… Offset voltage drift at the input of A to D (ADC) is given in terms of 1 to 5 micro volt per degree c. So, if it is 5 micro volt… For example, for 5 micro volt per degree c drift case, for delta T is equal to 100 degree c, then total drift is 500 micro volt.

Now, for (Refer Slide Time: 44:11) 16 bit converter, this error is equal to this drift, will be equal to 500 by 16; that would be 32 LSBs. So, the offset voltage drift also creates a problem. But, then there is similarly gain drift. The gain drift also will produce an error that one can calculate from the datasheet what will be the actually drift. So, one have to be aware that in most of the 8-bit converters, you can believe that resolution is 1 LSB and accuracy also is easily achievable to extent of 1 LSB. But, for 12 bit and 16 bit, one really have to calculate whether the accuracy and resolution are one and the same.

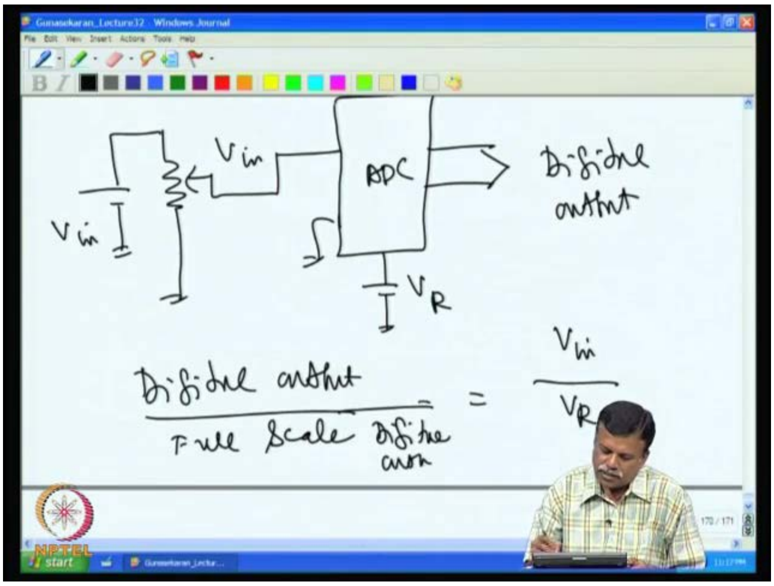
(Refer Slide Time: 45:06)



So, the lesson is that for 8-bit converters, resolution and accuracy will be more or less same less same; that is, it is possible to get 8-bit accuracy against ambient temperature variation. However, it is not possible to get this in the case of 16 bit. So, for 16-bit converters, it is difficult to get 16-bit accuracy. So, one has to really compute and see in the case of 12-bit and 16-bit converters, what the real accuracy is. So, one should not assume that resolution and accuracy are same in all the A to D and D to A converters. So, these errors are there in D to A converter also. D to A converter also you will have linearity error; then, you will have differential linearity error; you have offset voltage error, gain error, and then reference voltage drift error. So, these errors also can be computed in the same manner.

Now, we can look at it. What are the other possible techniques we can use to reduce the A to D converter errors? One common technique that is used is ratiometric conversion. In ratiometric conversion, we can avoid… Most of the temperature-related errors can be reduced when going for ratiometric conversion.

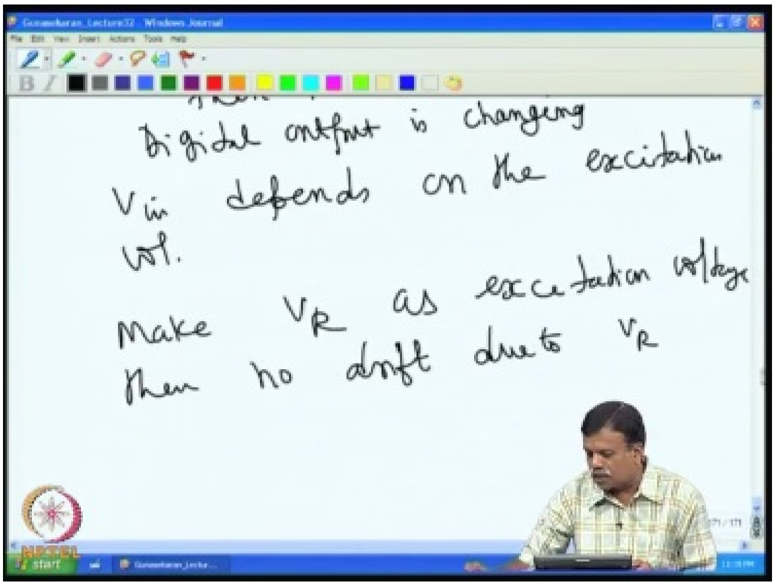
(Refer Slide Time: 47:56)



Let us see what the ratiometric A to D conversion is. Actually, there is no separate A to D converters. How to use the ratio metric conversion? I will not call this as a ratiometric ADC; I will call ratiometric conversion. This is only how to utilize the converter. So, we have… Let us take for example… If I take A to D converter, ADC is there; then, we have a reference voltage; and then, we have a V input. For example, I want to measure the resistance values accurately using for example, bridge circuit; or, I want to measure the resistance… potentiometer value to be measured, I take… For example, I can have a potentiometer like this; that is, I think is input; then, I give this as V input.

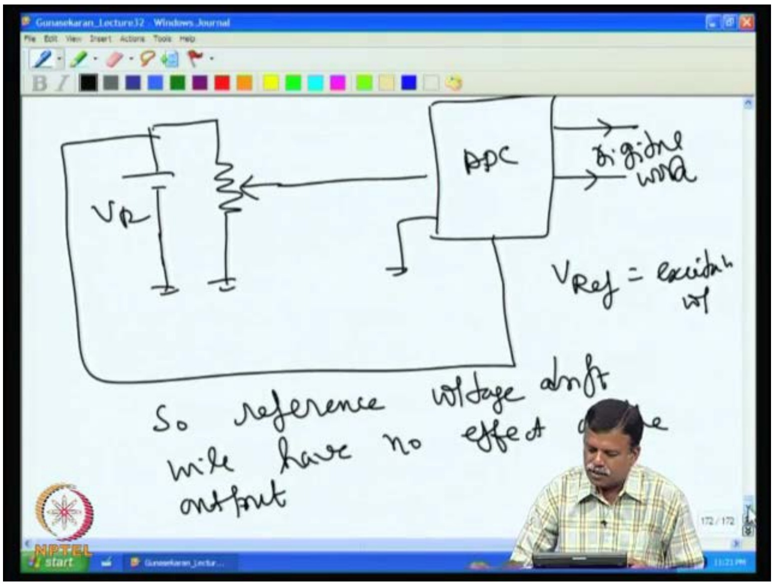
Now, I will get a digitized output here (Refer Slide Time: 49:03). This is V reference. Now, we know that in A to D converter, essentially, the digital value is nothing but the ratio between the two voltages; that is, what is the input and output? So, digital output divided by full scale digital output is actually equal to V in by V reference; that means in actual case, even V R changes with temperature, since this is already fixed. So, we will get that the digital output is changing when V input is not changing.

(Refer Slide Time: 50:03)



So, actually, if V R changes, then for the given V in, digital output is changing. Suppose in this case, if I put V in itself… This is actually not V in, V excitation (Refer Slide Time: 50:45). If I make… output is changing. Now, V in depends on the excitation voltage. One solution for this drift would be make V R as excitation voltage. Then, no drift due to V R; that is, what you do is, we will convert this excitation voltage and the references, are kept same.

(Refer Slide Time: 51:37)

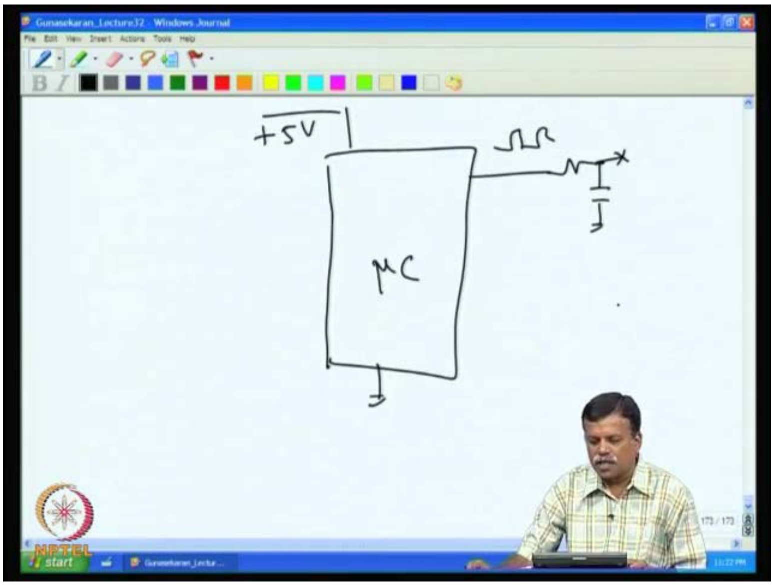


So, what you do is we will have the potentiometer here. For example, I want to measure a displacement; then, what will be the converter? We have a V reference here. This V reference is given to the A to D. Then, you have the input here and then output here. So, V reference is equal to excitation voltage. So, both are drifting. If the V reference drifts, then the reference to this also is changing; net result is it has no effect at the output digital; this is the digital word. So, reference voltage drift will have no effect on the output.

Now, of course other errors will be still there; that offset voltage drift and gain error – all that will be there. But, the reference drift, which is a major issue… Particularly for 16-bit converters, it is a major issue. That error can be removed by going for the ratiometric conversion. But, it is not possible all the time to go for ratiometric conversion. Wherever possible, one can go for the ratiometric conversion to avoid the error due to the reference voltage drift.

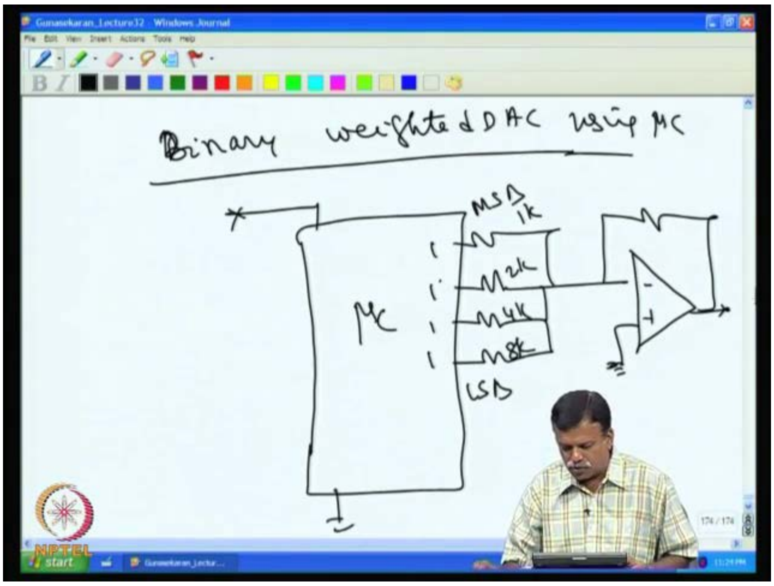
Now, we have (( )) based A to D converters, which are extensively used in the industry. So, one have to worry about what are these errors involved, because most of them believe that if it is an 8-bit converter, you will get 8-bit accuracy. That is no more true. So, one have to verify when you are using a (( )) based system, inbuilt ADC is there; one have to clearly verify from the data sheet what is the actual error associated with the particular converter.

(Refer Slide Time: 54:18)



Similarly, when you are using a D to A converter… One can also make a D to A converter (DAC) using micro controller. For example, in the case of microcontroller based system, we can also have a DAC. Which we already discussed that we have a plus voltage; for example, we can have an output, which is a p (( )); and then, we have an RC network, which gives you the analog voltage. But, then this is a very slow system. One can also make R to R DAC or binary weighted DAC using microcontroller very easily.

(Refer Slide Time: 54:51)



Binary weighted DAC using microcontroller – one can go for… For example, if you have a microcontroller, I can have a binary weighted converter; for example, I can have a… Resistance is connected to each one of them. For example, 4-bit converter if I want, then if all of them are one, for example, this is MSB and this is LSB, then I will have the error; I will have the output voltage corresponding to the digital word. Of course, the resistors are to be scaled correctly. For example, if this is 1 k, then the next one has to be 2 k, then 4 k, 8 k. This is one of the major problem that one has when you are dealing with microcontroller, because having to get these resistors correctly is the major issue. Then, immediately, you can realize if it is a 16 bit converter you want to realize, then it is even much more difficult, because you do not get the correct values. Of course, one can have… 4-bit converter can be easily achieved using this binary weighted D to A converter techniques, because getting a separate A to D and D to A converter is an expensive proposition; and, if it is a simple system, it is easier to use microcontroller itself for A to D and D to A function to save the cost.

So, we have discussed three A to D converters; that is, flash ADC, successive approximation ADC and dual slope ADC. And, we have also shown how D to A converters are working – both R to R network and binary weighted D to A converter; how to use them with microcontroller also we have discussed. So, we are left with one more A to D converter, that is, sigma delta A to D converter; that we will discuss in the next lecture.

Thank you.