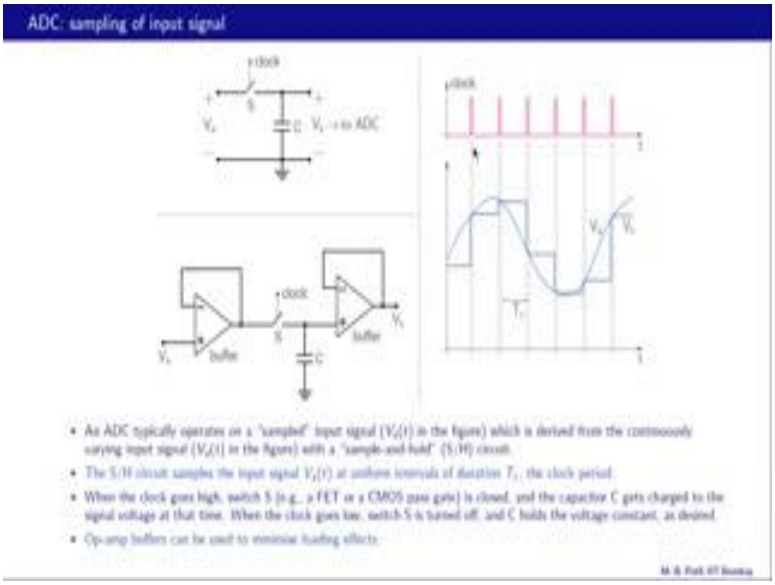
Basic Electronics
Prof. Mahesh Patil
Department of Electrical Engineering
Indian Institute of Technology, Bombay

Lecture – 72
Analog-to-digital conversion

Welcome back to Basic Electronics. We have looked at the flash ADC. In the last class we will now look at some other commonly used ADC types namely; successive approximation ADC, counting and tracking ADCs and dual slope ADC. Let us start.

(Refer Slide Time: 00:39)



Let us now discuss a very important issue which is very closely related to analog to digital conversion, and that is sampling of an analog signal. In many applications when we convert an analog signal into a digital signal we want to use digital signal processing on this digital output. And in these cases sampling is absolutely essential. So, let us first see what sampling means and then we will see how to achieve it.

Here is our analog signal the one that is varying continuously V a and they have a clock and the clock has very narrow pulses ideally with 0 width. Now what we want to do is when the clock is high we want to sample this analog signal and when the clock becomes slow we want to hold that sample value constant like that. When the next clock pulse comes along we want to now sample to analog voltage again and then hold that value

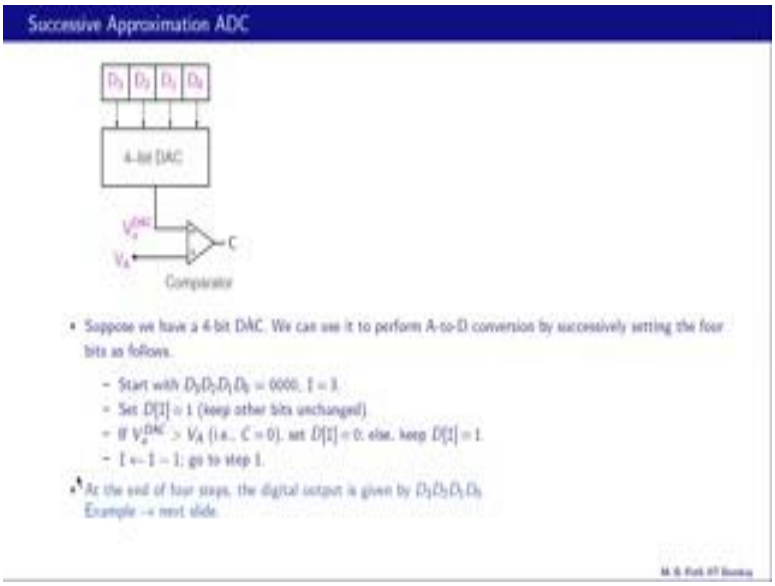
constant and so on. So, this process is called sample and hold. And these are the samples which are used for digital signal processing.

Now, digital signal processing also requires digital data, and therefore these samples need to be converted to a digital output and that we do using an ADC which is not shown in this figure. Here is a schematic representation of a circuit which can be used to perform the sample and hold operation. This is the input analog voltage and that is the sampled analog voltage which will be sent to an ADC for conversion to a digital number.

Now, this part is not shown in this figure, here is a switch which is controlled by this clock here. When the clock is high in a short interval here the switch closes and the capacitor gets charged to V a. Now this charging process is very fast because the switch has a very small resistance ideally 0, and therefore the capacitor voltage which is our sampled voltage simply follows V a of t and that is what we show over here. During this interval when the clock is low the switch is off and now the capacitor voltage cannot change anymore, because there is no path for the capacitor to charge or discharge. And that corresponds to this hold operation here. So, that is how this circuit performs the sample and volt operation.

Here is the summary of the points that we just made. In addition notice that the switch S is typically a field effect transistor or a CMOS pass gate this switch here. And in addition to this basic circuit we may need op-amp buffers both on the input side and also on the output side to minimize loading effects. So, this in a way is a more complete circuit.

(Refer Slide Time: 04:26)

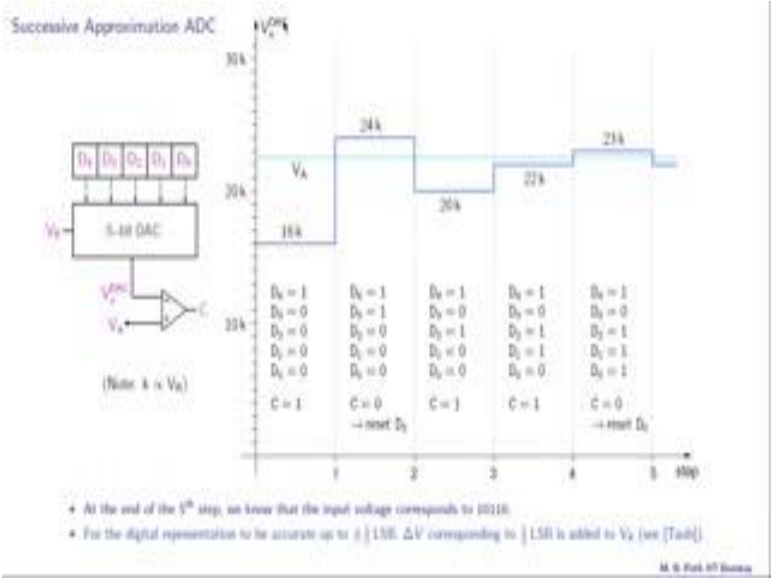


Another commonly used ADC is the successive approximation ADC; here is the block diagram for a 4-bit ADC. This register holds a binary number D 3, D 2, D 1, D 0; there is a 4-bit DAC here which converts this binary number to an analog value which we have called V o DAC output voltage of the DAC. That gets compared with V a which is the input analog voltage which we want to convert to the digital format. So, these are the operations involved and now let us see how this ADC works.

So, here is the algorithm we start with D 3, D 2, D 1, D 0 equal to 0 0 0 0; that means, all these bits are set to 0. And we define one integer I we said that to 3. Set D of I equal to 1 keep other bits unchanged; that means, since I is three we are going to set D 3 equal to 1 this bit here and keep the other bits D 2, D 1, D 0 unchanged that is they will stay at 0 0 0 in the first iteration. Now if V o back is greater than V a; that is if c is equal to 0 this variable here then set D of I equal to 0, else keep D of I equal to 1. And then follow this process by replacing I with I minus 1.

So, that is the algorithm and it will become more clear when we look at an example in the next slide. At the end of four steps the digital output is given by D 3, D 2, D 1, and D 0. So, at the end of four steps the contents of this register will be their digital output that we are looking for.

(Refer Slide Time: 06:50)



Let us now consider this example; it is a 5 bit ADC which uses a 5 bit DAC. And the output of the DAC which is V o DAC is plotted over here, and if this number is 0 0 0 0 0 then the output of the DAC is 0 volts here if this number is 1 1 1 1 1 that is decimal 31 then the output of the DAC is 31 times k that 1; where k is proportional to the reference voltage applied to the back. Our input analog voltage is shown by this line here, we will call that V a.

Now the algorithm begins like this: in the first step we set this m s b equal to 1 and all other bits equal to 0. And what is the DAC output then? This number is decimal 16, so 16 k is the DAC output as shown over here. Now, we compare this voltage with the input analog voltage using this comparator and we find that V a is higher. So, the comparator output C is 1 and therefore we keep this D 4 as 1 and then go to the next iteration.

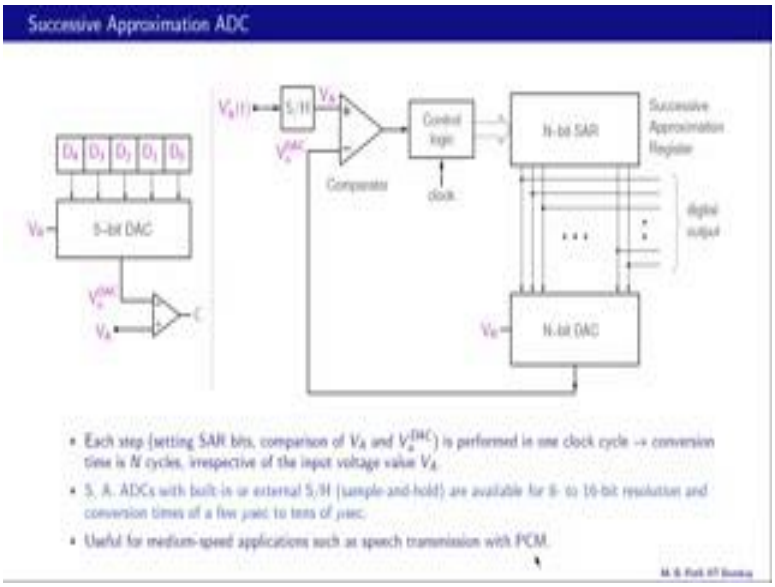
In the next iteration we set the next bit that is D 3 equal to 1 and once again compare V o DAC and V a. What is V o DAC in this case? It is 1 times 2 raised to 4 plus 1 times 2 raised to 3 that is 16 plus 8 for 24 k. And now we find that this V o DAC is higher than V a and that gives us c equal to 0. And since c is 0 what we do is reset the current bit that is D 3; that means make D 3 equal to 0 and then go to the next iteration. So, in the next iteration we have D 3 equal to 0 here.

And we continue like this, in the next step we make D 2 equal to 1 and we find our V o DAC is now 20 k V a is higher than V o DAC, C is 1 and therefore we retain D 2 equal

to 1 in the next step. Now, we set D 1 equal to 1 and find that V a is higher than V o DAC, so therefore v retained D 1 equal to 1. And finally, we make D 0 equal to 1 we find that our V a is now lower than the V o DAC and therefore we reset D 0. So, at the end of the fifth step you know that the input voltage corresponds to 1 0 1 1 0 that is 1 0 1 1 0, because D 0 has been reset.

One small correction is required for the digital representation to be accurate up to plus minus half LSB delta V corresponding to half LSB is added to V a; and you can read more about this in this reference.

(Refer Slide Time: 10:35)



Let us now look at how a successive approximation ADC can be implemented. This is the example that we considered in the last slide and this is how it can be implemented. We have an input voltage V a prime of t, a time varying analog voltage. And this voltage is fed to a sample and hold circuit to produce this V sub a; the same as this V a shown here. Now this V a is expected to be held constant throughout the conversion process, and therefore we must adjust the frequency for this sample and hold circuit the clock frequency so that V a remains constant throughout the conversion process.

So, that V a is one input of the comparator, the other input this one comes from the DAC and that we have called as V o DAC over there. The output of the comparator goes to this control logic which figures out which binary number should be loaded into the N-bit successive approximation register in the next clock cycle. This control logic gets this

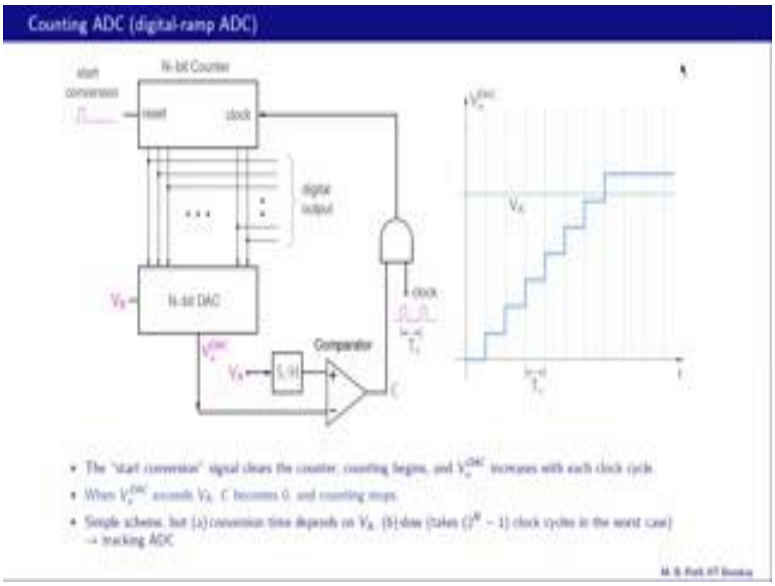
clock signal and that should not be confused with the clock signal for the sample and hold circuit. This clock signal essentially controls the speed of this conversion process.

As an example let us say that this clock has a frequency of 1 megahertz; that means the clock period is 1 microseconds. And let us say that it takes 20 cycles; 20 clock cycles to complete the conversion process. That means, this entire conversion from analog to digital is going to take 20 microseconds. And therefore, we must make sure that this V a is held constant for at least 20 microseconds. The output of the N-bit successive approximation register or SAR in short is sent to the DAC. And it also serves as our final digital output when the conversion process is completed. So, that is the overall block diagram of the implementation of the successive approximation ADC.

Let us make a few remarks each type that is setting of the SAR bits; these bits for example, and comparison of V a and V o DAC this comparison each performed in one clock cycle and now we are referring to this clock signal here. And therefore, the conversion time is N cycles. So, if this clock has a period of t then the total conversion time is n times t; where n is the number of bits. And that is irrespective of the input voltage value V a. Now this time the conversion time is definitely larger than that required for a flash ADC, but as we will see it is better than some other ADC types.

Commercially successive approximation ADCs with built in or external sample and hold are available for 8 to 16 bit resolution, and the conversion times are typically few microseconds to tens of microseconds. So, these are considered medium speed ADCs and they are useful for applications such as speech transmission with PCM for example. PCM is pulse code modulation.

(Refer Slide Time: 14:51)



Let us now discuss the counting ADC or also called the digital ramp ADC. Here is the block diagram: it has an N-bit counter with the reset input and a clock input, the output of the counter all these bits is given to N-bit DAC which converts it to an analog value which is called V o DAC over here; the output of the DAC. This is our input analog voltage that goes through a sample and hold circuit. And that is applied to the plus input of this comparator; the minus input is V o DAC. The output of the comparator c is ended with the clock signal and that serves as the clock for the counter.

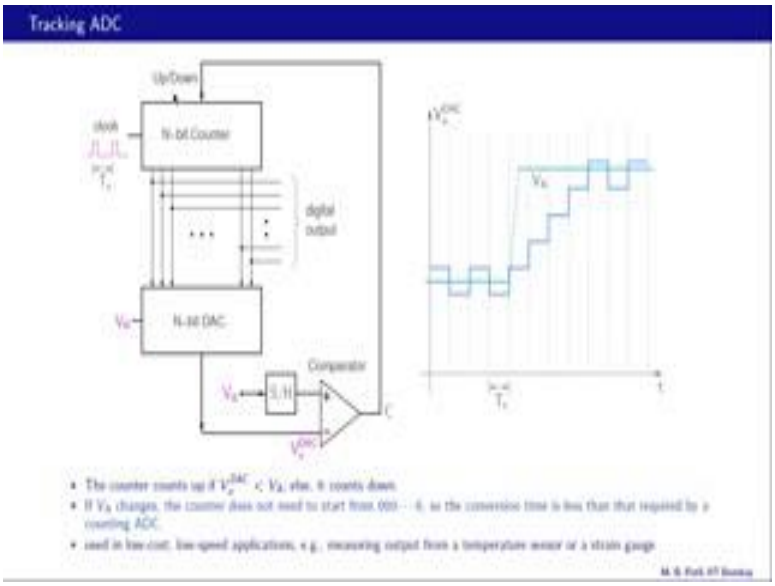
So that is block diagram; and now let us see how the ADC functions. The start conversion signal this pulse here clears the counter; that means, all these bits are reset to 0, counting begins and V o DAC increases with each clock cycle the output of the DAC here. And that V o DAC looks like this because the count is increasing with every clock cycle. Then V o DAC exceeds V a c, C becomes 0 and counting stops. Let us look at this point. This is our input analog voltage after sample and hold, and at this point our V o DAC exceeds V a. When that happens C becomes 0 and the output of this and gate therefore becomes 0; and the counter does not receive the clock so counting stops.

Now at this point the binary number that we have here in the counter is the digital output that we are looking for. If V a was larger for example somewhere here, then this counter would count for some more time and then stop. So, that is how this counting ADC works. So, this is a very simple and therefore attractive scheme, but a the conversion

time depends on V a as is obvious if V a is here then we need that many clock cycles, if V a is here then we need a larger number of clock cycles; b it is a slow process and in the worst case it takes 2 raise to n minus 1 clock cycles. So, if n is 10 for example, 10 bit ADC then this number is 10 24 minus 1 that is a large number of clock cycles.

And to improve upon the speed we have this ADC called the tracking ADC and that is what we will see now in the next slide.

(Refer Slide Time: 18:14)



Here is the block diagram of a tracking ADC, and it is quite similar to the counting ADC which we saw in the last slide. The only difference is that here we have an up down counter, and if you recall in the counting ADC the counter always started from all bits equal to 0 that is decimal 0; here that is not the case.

So let us see how this works. In this case the clock is always provided and whether the counter counts up or down is decided by this output see of the comparator. Now for the comparator V plus is the input analog voltage V a after sample and hold and V minus is the same as V o DAC the output of the N-bit DAC.

Let us take an example say V a is larger than V o DAC then C would be 1, and that signals the counter to count up then the count increases and V o DAC increases. So, in this manner V o DAC tries to catch up with the input analog voltage. In the other case when V a is less than V o DAC the comparator output is 0 and that makes the counter

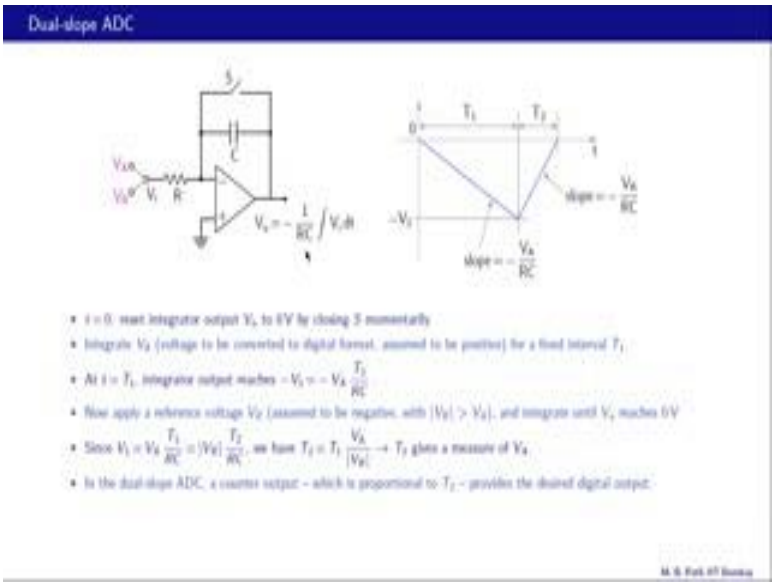
count down; that means the count decreases with time, and therefore the output of the DAC V o DAC will also decrease with time. So, once again we see that V o DAC is trying to catch up with V a.

In the steady state V o DAC oscillates around V a as shown over here, this is our V a and that is our V o DAC. Now in this clock period for example, V o DAC is higher than V a, so the counter will be asked to count down. So, the count has gone down. So, V o DAC has decreased by 1. And now we find that V o DAC is less than V a, so the counter will be asked to count up and so on. So, that gives rise to these oscillations around V a.

These oscillations may seem like a problem, but notice that the oscillations are only in the LSB, this least significant bit of the counter. And therefore, if we have a large enough number of bits where all these other bits remain constant then this change in the LSB may not really matter in a real application. Here is an example of what happens if V a changes from one level to another. In this case V a has increased, so the counter then starts counting up like that and finally when the counter catches up with V a that is V o DAC catches up with V a then the steady state is reached once again.

In summary the counter counts up if V o DAC it is less than V a else it counts down. If V a changes the counter does not need to start from all 0s. So, the conversion time is less than that required by counting ADC which we saw in the last slide. And this kind of ADC the tracking ADC is used in low cost low speed applications, for example measuring output from a temperature sensor or a strain gauge where quantities do not vary very fast; the time scale may be in the milliseconds range.

(Refer Slide Time: 22:14)



There is another useful ADC type that is the dual slope ADC and it is often used in handheld multi meters for example. The basic principle of a dual slope ADC can be illustrated with this circuit which is an integrator with V o equal to minus 1 over RC integral V i dt as we have seen earlier. A switch is provided in parallel with the capacitor. When the switch is closed V o is the same as V minus which is the same as V plus that is 0 volts, so when the switch is closed V o is forced to become 0. The input voltage for the integrator that is V i is either V a the input analog voltage or it is a reference voltage V R. And there is a switch which connects these two nodes or these two nodes.

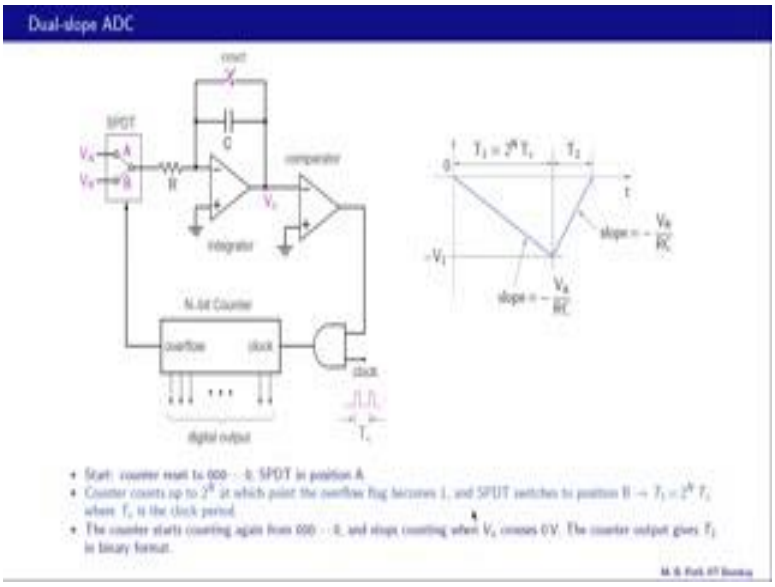
Let us now see how this ADC works. Let us say that at t equal to 0 we reset the integrator output to 0 volts by closing s momentarily, so we close this and open it very quickly. So, that makes the output voltage equal to 0 volt so that is our starting point. We should remark here that this switch is not a mechanical switch it is an electronic switch; for example the transistor and it is operated using some control logic which is inside the ADC chip. And it needs to close only long enough so that the capacitor gets discharged.

So, at t equal to 0 the switch has been closed and open and now the integrator can start integrating. In the first phase we put this switch in this position, so our V i is equal to V a that is the input analog voltage which we want to convert to the digital format. So, the integrator starts integrating V a and that happens for a fixed interval T 1. Now, let us assume that V a is positive and see what happens.

This is T 1 and since V a is positive and constant the output voltage is a straight line with time with a negative slope; that slope is minus V a by RC that comes from this equation. So, what is the value of V o at t equal to T 1? That is simply minus V a times T 1 by RC. And at this time that is t equal to T 1 we change the position of the switch from V a to V R, so V i is now equal to V R. And if V a is positive then we choose a V R which is negative and also larger than V a in magnitude. So, then what happens is V o starts increasing now, because V R is negative and this slope in magnitude is larger than this slope because our V R is larger in magnitude than V a. And this goes on until we reach 0. And let us say that we have some means of measuring this time interval T 2.

Now, from this figure we can conclude that V a times T 1 by RC is the same as mod V R times T 2 by RC, we have mod V R here because V R is a negative voltage. And from this equation we can say that T 2 is T 1 times V a by not V R. Now T 1 and mod V R are constants therefore T 2 is directly proportional to V a. In other words T 2 gives a measure of the input analog voltage V a, and that we convert into a digital format. So, in the dual slope ADC a counter output which is proportional to T 2 provides the desired digital output; and we will see how that is done.

(Refer Slide Time: 27:01)



Here is the actual implementation of a dual slope ADC. This part we have seen earlier that is the integrator, the output of the integrator V o is the same as V minus for this comparator it is V plus V is 0 and the comparator output is added with this clock signal

and output of the nand gate serves as the clock for this N-bit counter; this counter is n up counter binary up counter, and let us say it has 4 bits to illustrate what happens. It starts off at 0 0 0 0 then 0 0 0 1 etcetera and the final maximum count is 1 1 1 1. And after that it goes back to 0 0 0 0, but then it indicates that there is an overflow and that signal is used to change the position of this SPDT the single pole double throw switch from A to B.

So, let us go through the operation of the dual slope ADC now. We start with the counter with all bits equal to 0 decimal 0 and with the SPDT in position A; that means our V i for the integrator is equal to V a. So, the counter starts counting and the integrator starts integrating V o starts going down like that. The counter counts up to two raise to N at which point the overflow flag becomes 1; that is this point here. And at this point the SPDT switches to position B, and that control of course comes from this overflow flag. So, after this we have V i equal to V R.

Now, let us figure out what T 1 should be; we know that there are 2 raise to N clock cycles in this interval and therefore if T c is the clock period T 1 is 2 raise to N times T c. After this point the counter starts counting again from all bits equal to 0. The output of the integrator starts rising because our V R as the opposite sign, so V R is negative in this case. And when this V o crosses 0 what happens is the comparator output becomes 0 and we have 0 over here as the clock. So, this clock signal does not reach the counter and the counter stops counting. And the count that the counter holds at this time is exactly the digital output that we are looking for.

So, as we have seen in the last slide this T 2 is proportional to the input analog voltage V a and this counter simply converts that T 2 into the digital format. So, that is how this dual slope ADC works. It should be clear that the dual slope ADC is a slope ADC, its conversion time is T 1 plus T 2, and in the worst case T 2 can be nearly equal to T 1. So, that gives the total conversion time of 2 times T 1 or 2 raise to N plus 1 times T c where T c is the clock period.

Having said that let us also mention that the dual slope ADC has a major advantage in terms of noise immunity. Let us try to understand what that is. The analog voltage which we are trying to convert to the digital format is expected to be constant throughout the conversion process. However, there is some noise riding on it, and the noise can be both

positive and negative. When we integrate V a what happens to the noise because it is both positive and negative its integral is approximately 0 or a very small value and therefore it does not reflect in the digital output. And that is a big advantage of the dual slope ADC and that is why it is popular in applications such as handheld multi meters.

To summarize we have seen the successive approximation ADC counting and tracking ADCs and dual slope ADC. We have also commented on their advantages and limitations. That brings us to the end of this course, but we have only scratched the surface so far. It is important to remember that electronics is 10 percentage theory and 90 percentage practice. Then we actually want to build a product we need to know several practical matters such as; which op-amp meets the performance criteria, how to pick a diode or a BJT for a given application, how to design a PCB and so on, is there an alternative solution which may be better in some way.

This course will get you started, but you need to pick up many things on your own to become a practicing electronics engineer. Fortunately we have an excellent resource the internet, and there are some excellent books written by people who have worked in the area for a long time and built their own circuits, so you can make use of all of these resources. I wish you all the best in your careers, goodbye.