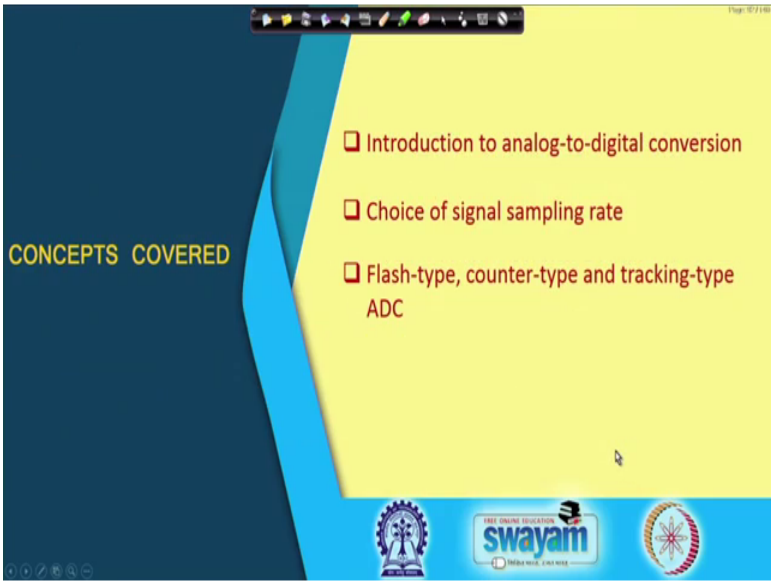
Embedded System Design with ARM
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 13
Analog to Digital Conversion (Part I)

In this lecture, we shall be starting our discussion on Analog to Digital Conversion. If you recall in our previous lecture we discuss the reverse process, how to convert a digital value in to an equivalent and proportional analog value this so called digital to analog conversion.

So, this lecture is titled analog to digital conversion the first part.

(Refer Slide Time: 00:42)

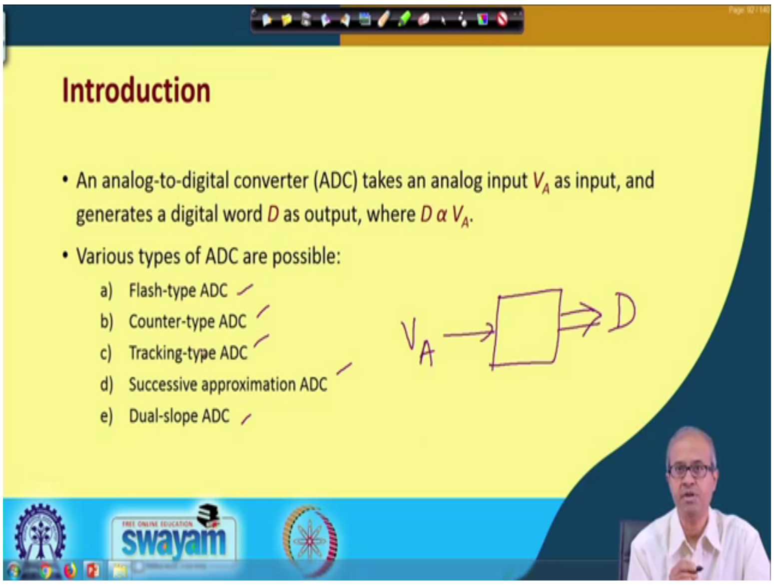


CONCEPTS COVERED

- ☐ Introduction to analog-to-digital conversion
- ☐ Choice of signal sampling rate
- ☐ Flash-type, counter-type and tracking-type ADC

And in this lecture we shall first be talking about some of the basics of AD conversion, then there is something called signal sampling rate, we shall be talking about that very briefly and we shall be looking at some alternate AD converter designs flash type, counter type and tracking type.

(Refer Slide Time: 01:06)



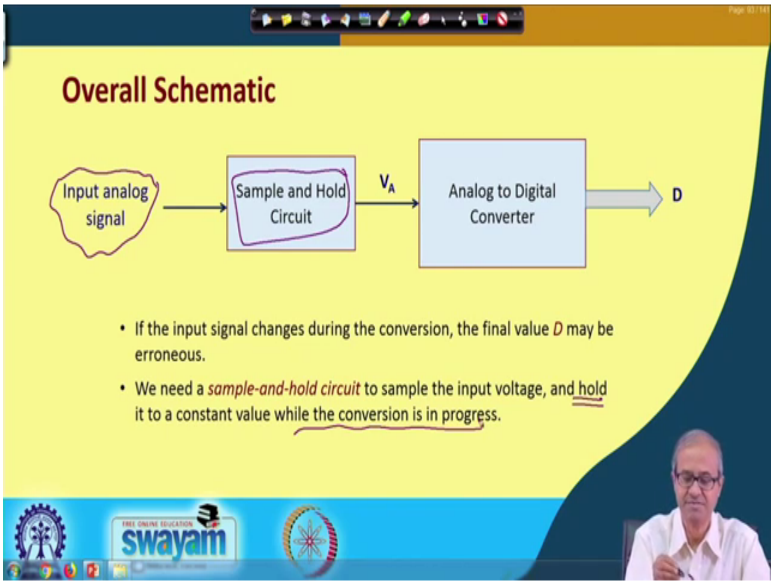
Introduction

- An analog-to-digital converter (ADC) takes an analog input V_A as input, and generates a digital word D as output, where $D \alpha V_A$.
- Various types of ADC are possible:
  - a) Flash-type ADC
  - b) Counter-type ADC
  - c) Tracking-type ADC
  - d) Successive approximation ADC
  - e) Dual-slope ADC

So, let us start with the basic introduction to AD converter and the various types that are possible. Now an analog to digital converter in short you call it an AD converter, it takes an analog voltage as input it takes as an input an analog voltage V A and it produces digital value at the output D, where D is proportional to V A ok.

Now, the number of bits in D can be different. So, if it is n in general we say that we have a n bit AD converter and various types of AD converter designs are possible some of them are listed here flash, counter, tracking, successive approximation and dual slope. During the lectures we shall be discussing the first four of these, the last one we shall not be discussing. The first four which are most commonly used we shall be discussing them ok.

(Refer Slide Time: 02:17)



Overall Schematic

Input analog signal → Sample and Hold Circuit → V_A → Analog to Digital Converter → D

- If the input signal changes during the conversion, the final value *D* may be erroneous.
- We need a *sample-and-hold circuit* to sample the input voltage, and hold it to a constant value while the conversion is in progress.

Before we start the discussion on the different types of AD conversion, let us look at the bigger picture what we are actually trying to achieve using this AD conversion. There is some analog signal which is coming from outside, this can be a voice, this can be a temperature waveform. Let us say, let us say in an industrial plant we are trying to implement some control circuitry, where we are monitoring the temperature pressure humidity of a oven or some plant.

So, those variables are being continuously monitor sensed. So, it will be like a waveform, changing up down up down it can change. So, it not necessary that they will they will remain constant and here you have the analog to digital converter which will be converting the voltage that is equivalent to the sensed value into a proportional digital output.

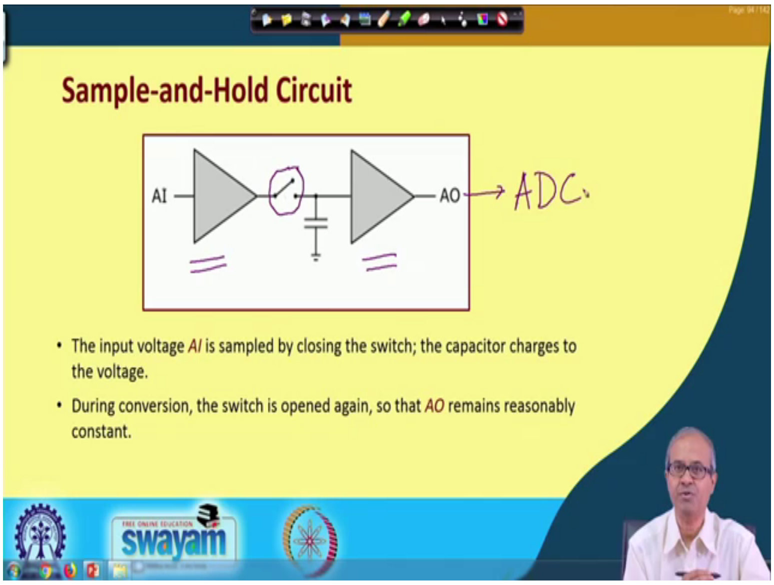
But you see you have another functional block out here which is mentioned called sample and hold circuit. So, why do we need sample and hold circuit? The reason is very simple, suppose my AD converter that I am using let us let us take it very typical example. Suppose it takes 5 milliseconds to complete 1 conversion, let us take an example 5 milliseconds. So, my input waveform is coming I am reading it and I am converting it the process takes 5 milliseconds.

Now, suppose my input waveform is changing very rapidly, now even within the five millisecond time the value of the input waveform is changing. So, what will happen?

During conversion if the input itself is changing. So, the final result may become erroneous. So, I expect that when AD converter is doing the calculation the input should be held at a stable value. So, sample and hold circuit does just that it samples the input at a particular time and holds it to that value while AD conversion is in process ok. This is the main purpose of sample and hold circuit.

So, sample and hold circuit will be sampling the input voltage and hold the voltage while the conversion is in progress. So, that the error in conversion is minimized.

(Refer Slide Time: 05:04)



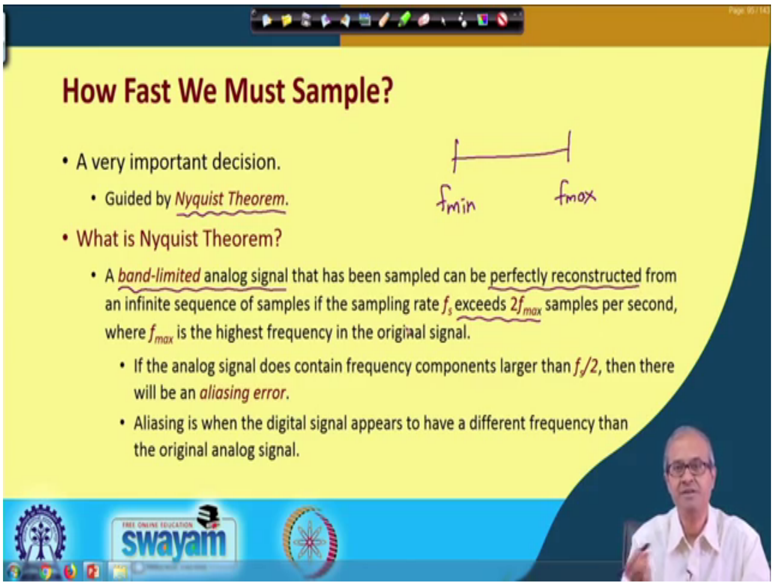
Sample-and-Hold Circuit

- The input voltage *AI* is sampled by closing the switch; the capacitor charges to the voltage.
- During conversion, the switch is opened again, so that *AO* remains reasonably constant.

So, I am not going into the detail circuit diagram what is that inside the sample hold circuit sample and hold circuit, but schematically I am just showing you what are the basic building blocks there. There are two buffers you can see, this is your analog input this is your analog output and there is a switch which can be controlled electronically, there is an electronically controlled switch like a relay, like a transistor there can be multiple ways.

So, whenever the switch is closed what happens is that whatever is this analog voltage this capacitor will get charge to that voltage ok. So, the switch will remain closed for certain time the time enough for the capacitor to get charged, then the switch is again opened. Now the capacitor does not have any path to discharge because this is a buffer, the input resistance of this buffer is typically very high. So, the capacitor cannot discharge. So, the output, analog output you can feed it to your AD converter. So, this voltage will be maintained at a stable level while AD conversion is in progress right this is how roughly a sample and hold circuit works ok.

(Refer Slide Time: 06:34)



How Fast We Must Sample?

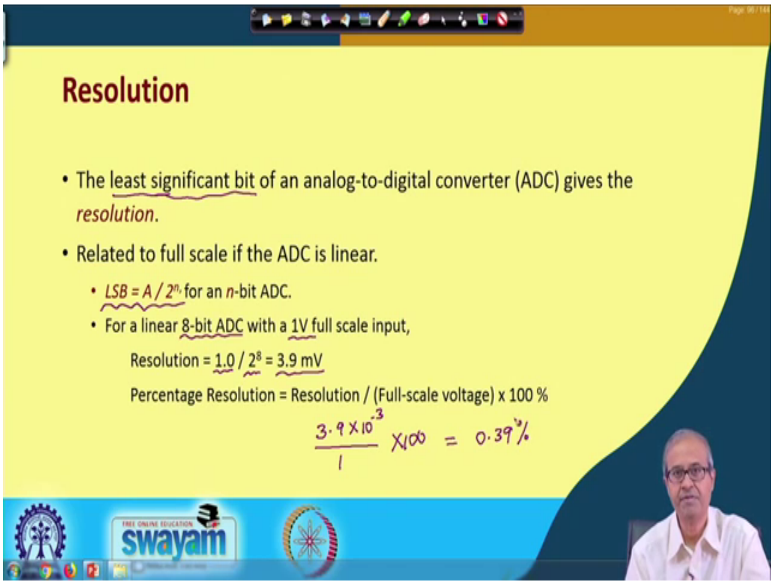
- A very important decision.
  - Guided by *Nyquist Theorem*.
- What is Nyquist Theorem?
  - A *band-limited* analog signal that has been sampled can be perfectly reconstructed from an infinite sequence of samples if the sampling rate f_s exceeds $2f_{max}$ samples per second, where f_{max} is the highest frequency in the original signal.
    - If the analog signal does contain frequency components larger than $f_s/2$, then there will be an *aliasing error*.
    - Aliasing is when the digital signal appears to have a different frequency than the original analog signal.

Now, the next question is what should be the rate of sampling? Should we sample once every second, once every millisecond, once every microsecond what should be the best sampling rate. Well sampling rate depends on the characteristic of the input signal and there is a nice theorem called Nyquist theorem that gives us a guideline in this regard and what Nyquist theorem says a band limited analog signal. Band limited analog signal means some signal which has frequency components within a range, let us say f min to f max, there is a range of frequencies, a band the signal frequencies are limited within that band that is called band limited signal.

This has been sampled suppose we are sampling it, this can be perfectly reconstructed at the receiving end provided we carry out the sampling at greater than twice of f max whatever is the maximum frequency component in your input waveform greater than twice that frequency we have to sampling that rate. If you do that, then at the receiving end you can reconstruct all the frequency components accurately this is what Nyquist theorem says and this gives you a guideline what should be the required sampling rate. This of course, depends on an input signal what is the value of f max, it depends on f max fine.

But if you do not satisfy this condition greater than twice of f max, then your receiving end might get wrong you gets inferencing regarding the frequency components present in the signal ok. I am not going into detail, but this is the basic idea on the result of sampling.

(Refer Slide Time: 08:55)



Resolution

- The least significant bit of an analog-to-digital converter (ADC) gives the *resolution*.
- Related to full scale if the ADC is linear.
  - $LSB = A / 2^n$ for an *n*-bit ADC.
  - For a linear 8-bit ADC with a 1V full scale input,

Resolution = $1.0 / 2^8 = 3.9$ mV

Percentage Resolution = Resolution / (Full-scale voltage) x 100 %

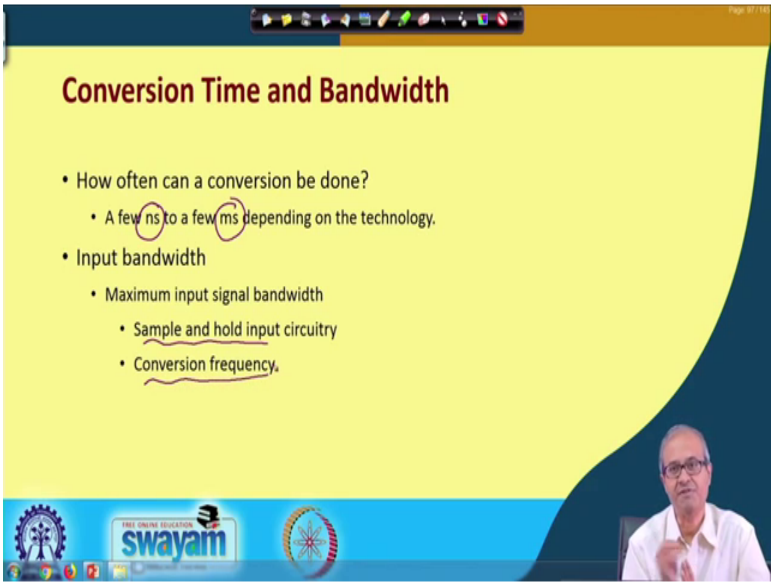
$$\frac{3.9 \times 10^{-3}}{1} \times 100 = 0.39\%$$

Now, this another thing called resolution of an AD converter, just like a DA converter we had resolution for AD converter also we have resolution this is the least significant bit. The least significant bit of an AD converter like the input is analog output is digital what is the minimum voltage I must change in the input so that in the output the least significant bit changes. That means, the output digital word changes by 1, this is like that step size I in DA converter, minimum change in the analog input which will result in a change in the output digital by plus 1 that is resolution.

Now, if the characteristic of the AD converter is linear that it changes linearly with changes in the input output, changes linearly then the resolution of the LSB will be defined as A, A is the full scale voltage divided by 2 to the power n for an n bit DA converter.

So, let us take an example suppose I have an 8 bit AD converter and my full scale voltage is 1 volt let us say. So, my resolution will be full scale voltage 1 volt divided by 2 to the power 8. So, if you calculate it comes to 3.9 millivolts and this can be expressed also as a percentage that mean this resolution whatever you have calculated 3.9 millivolt divided by the full scale voltage; so, whatever full scale voltage you have multiplied by 100. So, in this case what will be it will be 3.9 millivolts; that means, 3.9 into 10 to the power minus 3 so many volts divided by full scale voltage is 1 volt. So, both are in volt, they will cancel out into 100. So, if you 10 to the power minus 3 into 100 this will be 0.39 percent right. So, in this way you can calculate resolution and percentage resolution.

(Refer Slide Time: 11:18)



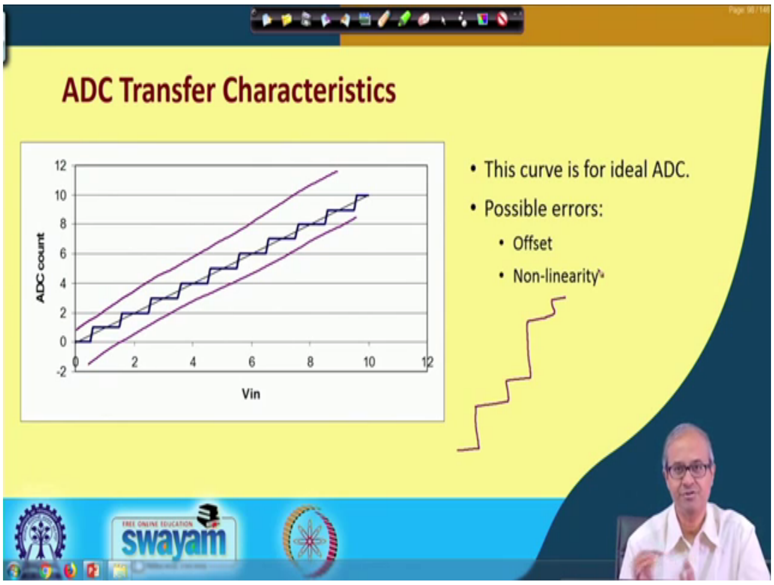
Conversion Time and Bandwidth

- How often can a conversion be done?
  - A few ns to a few ms depending on the technology.
- Input bandwidth
  - Maximum input signal bandwidth
    - Sample and hold input circuitry
    - Conversion frequency.

Now conversion time and bandwidth are two other parameters, conversion time is how much time it takes for the AD converter to convert a convert a given input analog voltage into the output, digital equivalent right.

So, depending on the type of converter it can be very fast, it can take few nanoseconds it can be quite slow also milliseconds or of the order of milliseconds, it depends on a type of converter right. And input bandwidth is that what is the maximum frequency of the input that can be supported of course, it depends on two different component the circuit. One is how fast is your sample and hold circuit and how fast is your AD converter, both we will determine that maximum frequency alright.

(Refer Slide Time: 12:17)



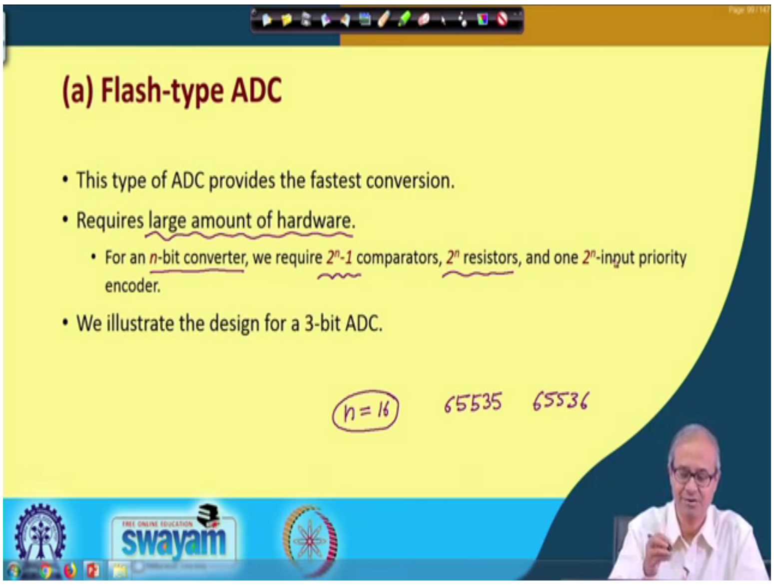
ADC Transfer Characteristics

- This curve is for ideal ADC.
- Possible errors:
  - Offset
  - Non-linearity

Talking of the ADC transfer characteristics. So, as the input changes this is the digital output 0, 1, 2, 3, 4 there is also there will be a stepwise increase just like a DA converter. It goes up like this, this curve is for ideal case because this is a straight line, but in general that can be offset error or nonlinearity error. Well offset error means instead of this you may see that your wave your waveform is like this either like this or like this.

Nonlinearity means the step height may not all be equal, for one step you may see that it is going big second step may be small, third step may be big fourth step may be small this is called nonlinearity. So, such errors may be there in an actual implementation fine.

(Refer Slide Time: 13:13)



(a) Flash-type ADC

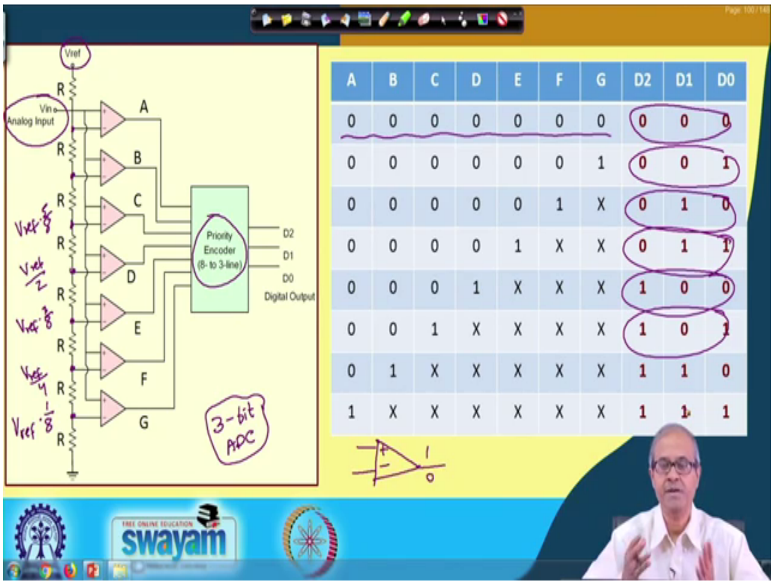
- This type of ADC provides the fastest conversion.
- Requires large amount of hardware.
  - For an n-bit converter, we require 2^n-1 comparators, 2^n resistors, and one 2^n-input priority encoder.
- We illustrate the design for a 3-bit ADC.

$n = 16$ 65535 65536

Now let us come to the different types of AD converter and how they work, the first type of AD converter that you talk about this is called flash type AD converter. Well this has nothing to do with flash memory flash means very fast, this is the fastest type of AD conversion that is possible that is why it has been given name flash AD converter. But the drawback of this converter is that it requires enormous amount of hardware, how large? Let us I have an estimate here suppose we want to design an n bit AD converter.

Let us say n equal to 16 let us say n equal to 16 I want a 16 bit AD converter for first application. I will be needing 2 to the power n minus 1 comparators, how much will be 2 to the power n minus 1 if any 16 it will be 65535. So, I need 65000 comparators, just imagine 2 to the power n resistances I need 65536 resistances so many resistances and one priority encoder that will be having 65536 inputs, huge size priority encoder. So, which is impractical to build so because of this characteristics you can only built very small flash AD converter for small values of n, larger values of n it becomes really impractical ok. Let us see how flash AD converter looks like and how it works ok.

(Refer Slide Time: 15:09)



A	B	C	D	E	F	G	D2	D1	D0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	1
0	0	0	0	0	1	X	0	1	0
0	0	0	0	1	X	X	0	1	1
0	0	0	1	X	X	X	1	0	0
0	0	1	X	X	X	X	1	0	1
0	1	X	X	X	X	X	1	1	0
1	X	X	X	X	X	X	1	1	1

Here on the left hand side I am showing the diagram of a 3-bit AD converter flash type. Let us see how it works well in the first stage you can see that there is a reference voltage V ref that you have used and there is a chain of resistances to ground. So, what this resistances serve the purpose is that it provides several reference voltages let us see the lowest one.

Lowest one have a resistance R below and 1, 2, 3, 4, 5, 6, 7 7 R above. So, what will be the voltage here? V ref R r divided by 8 R; that means, 1 by 8. So, this will give a voltage of V ref by 8, if you move to the next one this will be a 2 R and this is 6 R. So, 2 R divided by 2 R plus 6 R. So, it is 2 by 8 one fourth this will be V ref by 4, then you look at this is 3 this is a 8 this will be a V ref into 3 by 8 this will be a V ref by 2 this will be 5 by 8 V ref into 5 by 8, this will be V ref into 2 by 3 and so on.

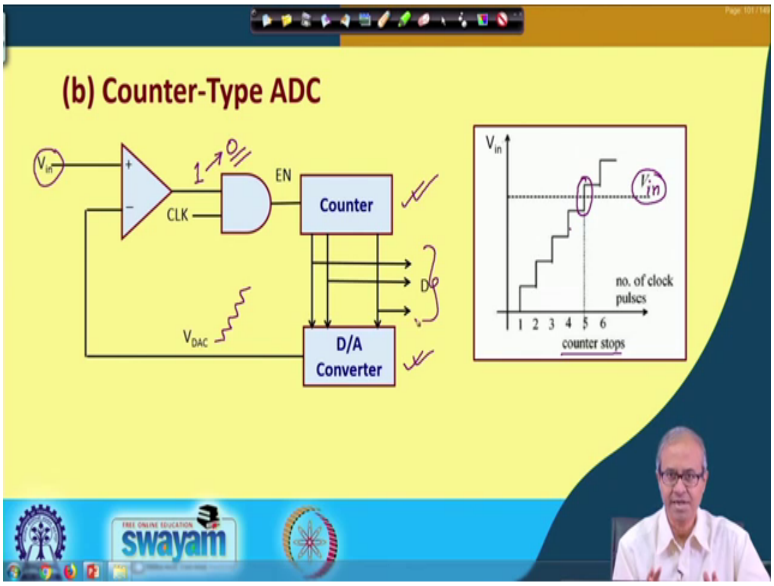
So, you see it provides a range of a reference voltages equally separated, here the difference of this voltages if you count is V ref by 8, V ref by 8, twice V ref by 8, 3 times V ref by 8, 4 times V ref by 8, 5 times V ref by 8 and so on and there are a series of voltage comparators. Voltage comparators how do they work? Suppose I have a comparator like this plus and minus, I have two inputs I apply two voltages.

If the plus input is greater than the minus input then the output is a digital output, output will be a 1 if minus is greater output will be 0 this is how comparator works. So, depending on what my analog input voltages V in is applied in parallel to all the plus inputs and this reference voltage are applied to the minus inputs. So, depending on your input voltage this A B C D E F G values will change.

So, you see if it is a very small voltage even smaller than V ref by 8 then all of A B C D E F G will be 0, the first row of this table. So, if it is between V ref by 8 and V ref by 4 here then only G will be 1 all others will be 0 and if it is in between the next two then the last 2 will be 1, last 3 will be 1, last 4 will be 1 like this and the this priority encoder will be converting these binary combinations, these inputs into equivalent digital output values.

So, what we get in the output will be the required ADC output digital value, here everything is done in a one go, the delay will be the delay of a comparator plus delay of the priority encoder that is it. There is no iteration or a loops anywhere, that is why this kind of converter is very fast, but as I told you it requires enormous amount of hardware alright.

(Refer Slide Time: 19:23)



(b) Counter-Type ADC

Let us look into something which is more practical to built you see DA converter is easier in that respect, you need only resistances you can built a 10 bit, 12 bit, 16 bit, 32 bit DA converter without any trouble right.

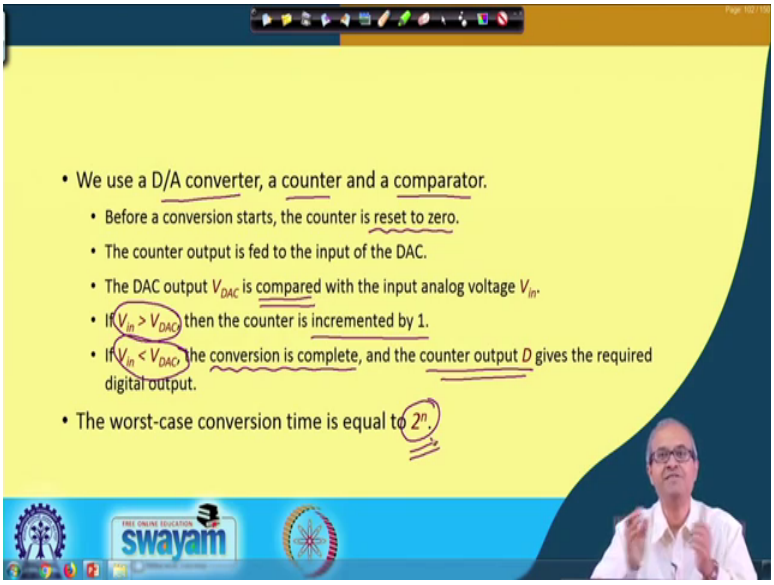
So, that is our assumption, we can built large DA converters. So, the methods that we talk about now there we have a DA converter as our central you can say block using that we are trying to realize an AD converter ok. This method is called counter type AD converter, just you look at the block diagram. So, we have a DA converter out here, the input of the DA converter is fed from a binary counter it counts 0, 1, 2, 3, 4 like that and this counter is not counting always, it will be counting only when this clock will be coming.

And this clock will be coming provide at the other input is at 1, this is an AND gate if the other input is 0 then the clock will not come. So, as long as the other input is 1 this counter will go on counting 0, 1, 3, 4. Now you tell me when this output will become 0? So, as the counter is counting up you think of this V DAC. What will happen to the output of the DA converter? So, it will also increase a slowly in steps.

So, as the counter increases so the minus input of this comparator will go on increasing and there will be a point when this minus input will be crossing your analog input voltage. So, as soon as it crosses this value will become 0 and the counter will stop counting ok. So, you see in this diagram we have showed exactly this thing, suppose this is our V in input voltage that you are applied this dotted line and the counter will starts from 0 it will continuously go on incrementing and as soon as it crosses and the counter will stop.

And whatever is the counter value that will be the corresponding digital output. This is how you are implementing an AD converter using a DA converter, reverse of that DA converter right this is called counter type AD converter because you are using a binary counter here.

(Refer Slide Time: 22:12)



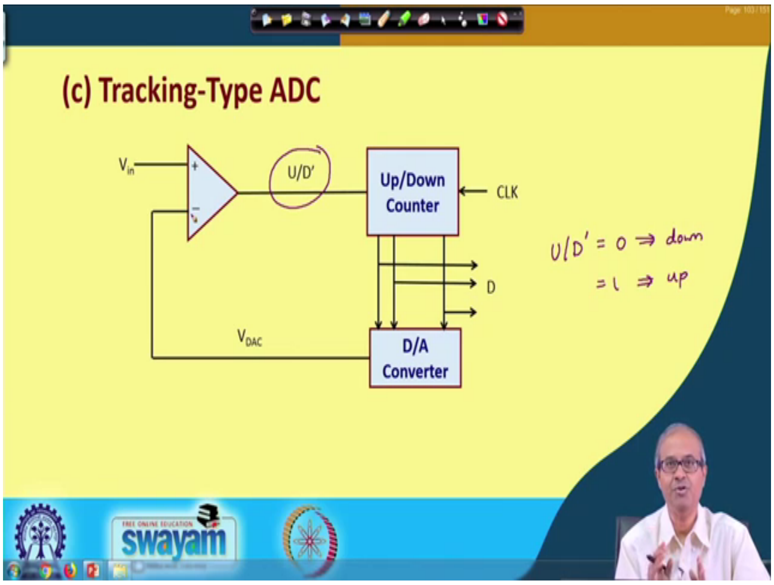
- We use a D/A converter, a counter and a comparator.
  - Before a conversion starts, the counter is reset to zero.
  - The counter output is fed to the input of the DAC.
  - The DAC output V_{DAC} is compared with the input analog voltage V_{in}.
  - If $V_{in} > V_{DAC}$ then the counter is incremented by 1.
  - If $V_{in} < V_{DAC}$ the conversion is complete, and the counter output D gives the required digital output.
- The worst-case conversion time is equal to 2^n.

Now, to summarize exactly what you are doing here the steps are written just I am reiterating. So, you need a DA converter, you need a counter you need a voltage comparator before the conversion starts the counter is reset to 0 and as you we have seen the output of the counter goes to the input of the DA converter ok.

Now, the DA converter output is fed to the minus input of the comparator and the analog voltage input is connected to the plus input of the comparator. So, we are making a comparison, if you find that the input voltage is greater than the DAC output then the counter is incremented by 1; that means, the and gate is enabled. But as soon as the reverse happens the V output of the DA converter crosses V in this will mean that the conversion is complete we stop and whatever is the counter output that will be the required output of the AD converter.

Now, one problem with this method is that in the worst case what will be the conversion time for an n bit counter I may have to count through all 2 to the power n steps to find where V in is. So, the maximum number of clock pulses required maybe 2 to the power n. So, the time is determined by the number of clock pulses, it can be 2 to the power n ok, this is one drawback here it is slow, but it is very simple we can built very large converters fine.

(Refer Slide Time: 24:00)



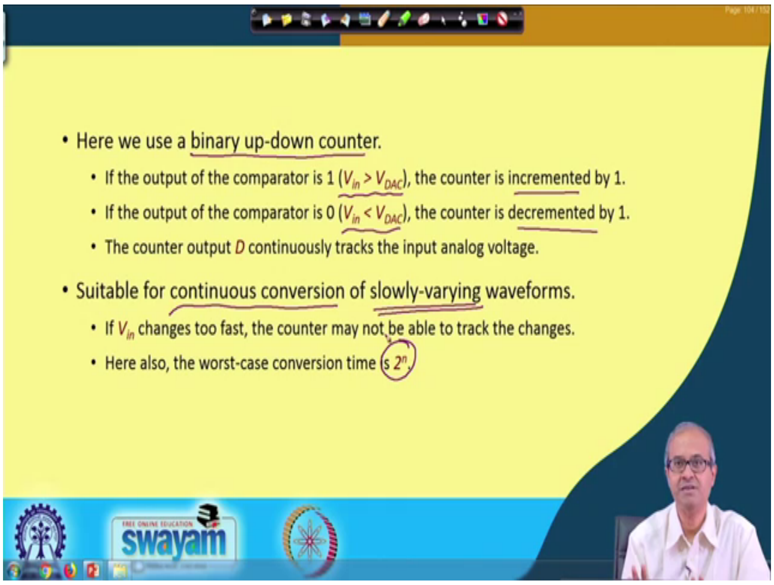
(c) Tracking-Type ADC

U/D' = 0 ⇒ down
= 1 ⇒ up

Now with the small modification you can make an improvement, while here we have something called tracking type AD converter you see this circuit looks quite similar, but we have remove that AND gate, but what we have done we have replaced this simple binary up counter by an up down counter. Up down counter is a binary counter which can either count up 0, 1, 2, 3, 4 or it can count down 4, 3, 2, 1. And how we decide whether it is going to count up or down? There is a control input called a up down bar, if this U D bar input is 0 this means the counter will count down, if it is a 1 this will mean the counter will count up ok.

Now, see here also the principle is very similar, you start by initializing the counter to 0 and then the comparator will be comparing DA converter output in V in, it is very simple if the input is greater than this U D the output will be 1. So, the counter will counting up, if DA converter becomes greater the counter will be counting down. So, the DA converter output will try to track the input voltage here you may assume that there is no sample and hold circuit. The input waveform is also changing slowly and DA converter output is trying to track it, it is a either increasing or decreasing depending on which direction the input is changing and this is happening continuously this is how tracking AD converter works.

(Refer Slide Time: 26:00)

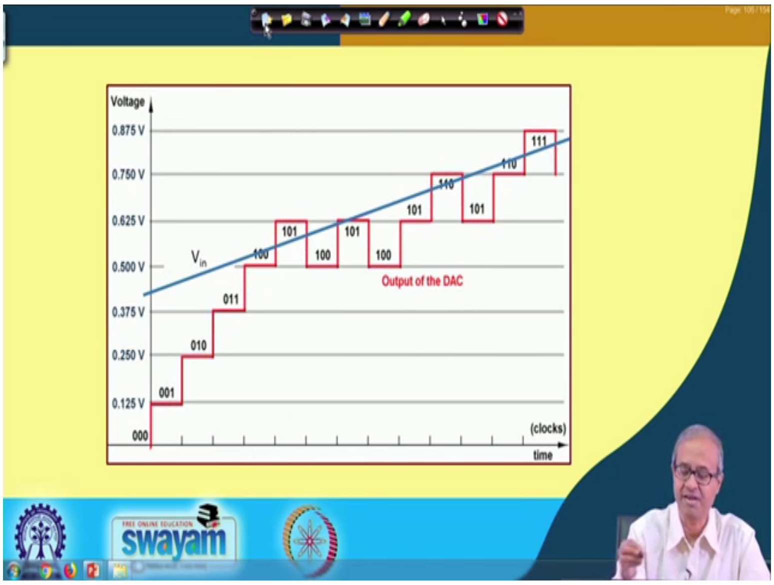


- Here we use a binary up-down counter.
  - If the output of the comparator is 1 ($V_{in} > V_{DAC}$), the counter is incremented by 1.
  - If the output of the comparator is 0 ($V_{in} < V_{DAC}$), the counter is decremented by 1.
  - The counter output D continuously tracks the input analog voltage.
- Suitable for continuous conversion of slowly-varying waveforms.
  - If V_{in} changes too fast, the counter may not be able to track the changes.
  - Here also, the worst-case conversion time is 2^n.

Now here again the steps we use a binary up down counter here and if the output of the comparator is 1; that means, V in is greater than V DAC we increment the counter, if it is reverse we decrement the counter. So, here there is nothing like conversion is complete we are continuously doing the conversion we are tracking the input voltage. So, for applications where the input is changing very slowly and we will need to continuously track the input there you can use this kind of AD converter ok.

So, this is suitable for a continuous conversion of very slowly varying waveform, but the worst case conversion is still 2 to the power n because if the input is very high you will have to count through all 2 to the power n steps to reach V in ok. So, that worst case delays complexity you are not improving here.

(Refer Slide Time: 27:09)



I am just to show how the input can track, suppose this blue line that I am showing, this blue line let us say this is my input waveform, my input waveform input voltages slowly changing this is my V in.

And this red one is the output of the DA converter which is trying to track. So, you see how it changes, it is going up it is out here it is crossing. So, the next step it will be counting down. Next step it is seen that again V in is up again it will go up, again it will go down, again it will go up, again up, down, up, up. And you see the DAC output is trying to track the input waveform as it is changing ok, this is the main characteristic of the tracking type AD converter.

So, with this we come to the end of this lecture. So, here we have discussed three different designs of AD conversion flash type, counter type and tracking type. In the next lecture we shall be talking about another type of AD converter which in fact, is faster than the counter type or the tracking type. In the sense that for n bit conversion you require only n number of clock pulses and which is very much practical; for a 16 bit AD converter you need no more than 16 clock pulses rather than 65000 clock pulses. This we shall be discussing in our next lecture.

Thank you.