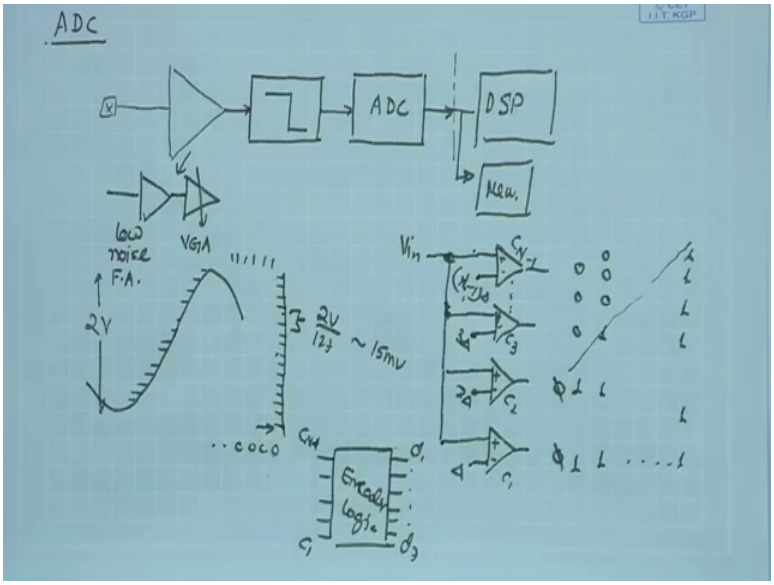
Analog Circuits and Systems through SPICE Simulation
Prof. Mrigank Sharad
Department of Electronics and Electrical Communication Engineering
Indian Institute of Technology, Kharagpur

Lecture - 36
ADC Introduction

Welcome to today's session. We are going to discuss the concept of analog to digital converter into today's session. And that is an overall, in the overall system that is an important part that is the last part in our analog signal processing change.

(Refer Slide Time: 00:35)



So, let us look at the system with which we began our discussion. We have the front end amplifier for which we have already had detail discussion which is acquiring the data from the sensor and we have seen the design issues associated with the front end amplifier. So, starting from higher level specification going down to the blocks and the circuit level specification transistor level, design, sizing, addressing different issues starting from noise, bandwidth, gain, linearity, precession and so on. Likewise we have discussion on the low pass filter.

So, the anti leasing filter which is suppose to reject high frequency noise and there we saw that we can use some special techniques like Gm-C filter for implementing very low frequency cutoffs and finally, we have the processed amplified filter signal which is going to the a to d converter over here. And next it goes to the digital domain and you

can do all your signal processing in the digital domain the digital signal processing that you want to do. You can also store your data in the digital memory conveniently. So, this is the overall system. And we have looked into the details of the first two and you are going to start with the implementation of the ADC.

So, we will first discuss the basic concepts of the ADC. Many of the specification of the ADC, we will skip at this point because the purpose of this course is to give you a design example. So, some of the definitions of, specifications of ADC will be skipping those in favor of time. And will be looking at the circuit implementation of simple relatively simple topology. We will try to start from high level specification of the ADC and it is building blocks given the signal conditions and then will be going into the design issues transistor level, block level, designing issues, dealing with the intricacies and the non idealities at transistor level design of each of those blocks.

So, that is going to be the approach. So, let us start with the high level description of the a to d operation that we are going to have on our signal. So, first of all relate regarding the swing we have already discussed that the overall amplification and filtering stage you suppose to give an overall gain, such that the weak signal over here is sufficiently amplified and it is map to the input dynamic range of the ADC. What is the input dynamic range of the ADC? That is basically the maximum peak to peak signal swing that the a to d converter can process. And in order to take advantage of the maximum dynamic range we always try that the signal coming after the analog processing is almost is peak to peak range is almost match to the input dynamic range of the ADC.

Now, to do that we have also seen what are the steps to we take in the front end amplifier generally. The front end amplifier is composed of at least 2 stages where you have the low noise amplifier the first stage where the noise consideration is very important. And the second one is a variable gain amplifier or VGA, I can call this a low noise front end amplifier and second one is the variable gain amplifier.

So, depending upon the current signal magnitude I can change the gain of the variable gain amplifier, we have seen how to program the gain of the VGA using programmable capacitive banks. Depending upon the signal magnitude at a particular time the DSP can send some feedback signal to the VGA some digital control signal to the VGA and adjusts it is gain. So, that at any time you just signal amplitude is say reducing the VGA

gain will be increased and vice versa, so that the overall magnitude of the signal coming over here is mapped to the input dynamic range of the ADC.

What we have seen in our previous examples, if that the VGA over here is capable of giving us almost full swing. And the for our 2 volt supply the swing can be almost close to 0 to V DD. And in general it may be beneficial to apply the VGA after the filter, So that the circuit implementing the filter did on suffer from excessive non-linearity. Because if you give a fully amplified signal to the filter fully input swing to the filter that can introduce to a lot of distortion and non-linearity in the filter itself.

So, it a better technique could be to inter pose the filter between the low noise front end amplifier and the VGA so that after the filter you and use the VGA an obtain a full swing signal which is going to the ADC. That is the another concept we have seeing and now, once we have this signal traversing this entire front end and after being process and amplified it is available at the ADC. We expecting that the swing of the signal over here is almost from 0 to VDT and therefore, the ADC over here also suppose except that entire signal. What is the basic function of an ADC is supposed to quantize? A continues time continues amplitude signal into n number of discreet levels.

So, it is supposed to quantize the impair input signal into n discreet levels. And generate a digital world which diplex the signal level with respect to this scale. For example, if we have say almost close 2 volt peak to peak signal and we want a 7 bit ADC for instance. So, we have been talking about the revolution and in the very beginning we have said that let us have at least one percent procession for the signal processing in the analog front in, So that close to 7 bit procession can be maintained.

Suppose we are going for 7 bit ADC and therefore, we are going to have 127 different levels over here or 127 different comparison levels over here. So, if I look at the individual section the different between the 2 levels is going to be around 2 volt upon 127 s it is comes around fifty mill volt. So, basically this entire signal which can range from 0 to 2 volt is divided into this 15 mill volt segment. And the ADC supposed to figure out that the current sample where is the current sample located in this entire range. And correspondences that it is suppose generate a digital world which in codes is this going to 27 levels.

So, we know that for 127 we need 7 bit digital world if the data is lower than the lowest reference, then we can say that all the 7 bit is will be 0 and if it is higher than the highest reference than all the 7 bit is will be 1 and in between we can accordingly have the corresponding analog value converted into the digital form.

So, what is the simplest way in which this can be done? If we just logically think of the overall operation overall functionality that we are trying to achieve the simplest operation is that you compares input signal base this thresholds, starting from 15 mill volt 30 mill volt 45 mill volt and so on. And then determine where what is the level of the signal with respect to all these threshold.

So, I can have bank of comparators and analog comparator if I talk about a idle analog comparator it is operation is well understood, if the if there is the reference signal and you are having an input signal coming over here and suppose you have a reference signal coming at the negative terminal of the comparator. If the signal is higher than the reference signal the output should be going high, a comparator is having only 2 output levels it can be either high or low.

So, generally if you are having an analog circuitry implementing of a comparator the high level corresponds to the supply which can be V DD and the low level is the 0 or the ground supply. So, these are the 2 possible level for a idle comparator high or low input can be analog we have reference which is the fix value which are trying to compare the input. And if the input is higher than the reference output will be V DD and if it is lower than the reference output will be 0 that is the function of a comparator.

So, in order to implement the of this quantization this simply quantization I can imagine having a bunch of comparators, which are or receiving the input signals on their positive terminal and they are having reference signals, reference voltages apply to the negative terminals. So, suppose this is my first comparator C 1 this is C 2 this is C 3 and likewise you have C n. And in order to have 7 bit quantization we will required 127 different comparator So that it can decide which of the 128 suppose the data is allocated to.

So, here I can have the V ref if I call this the lowest the difference between the reference level if I call this delta, the lowest level over will be delta which is 15 mill volts this will be 2 delta 3 delta and so on. And here if I call this n minus 1 delta then I can say that if the input signal is coming over here and these are the output of this comparator, if the

input signal is lower than the delta that is the lowest reference point. All the bit is will be 0 and all the comparator outputs will be 0 because in that case the signal coming at the positive terminal for all the comparator it is lower than the references.

So, it will get straight 0 outputs over here. And at the signal goes up suppose there is higher than 2 delta. But lower than 3 delta in that case it will have the first 2 comparator giving us 0 0 whereas, since it is lower than 3 delta I will have I am sorry, I will be getting a 1 and 1 at the output of the 2 comparator if the signal is higher than the 2 delta and other 2 comparators all the other comparators will be giving as 0 0. And likewise suppose the signal is higher than even 3 delta than I have 1, 1, 1. But if it is lower than 4 delta. So, the fourth comparator onwards all the value will be 0.

So, basically as signal goes up we are having some kind of thermometer code where the lower comparators outputs are going to we set to 1. And the upper comparators there will be 0 and if the signal amplitude increases larger number of comparators from the bottom point will be having output 1 and smaller numbers will be having output 0. So, the number of comparator having high output basically denote the signal level, if larger number of comparator starting from the lowest one are having once. That means, the signal is towards the higher value and the extreme cases when signal is as I said very close to the or even higher than the upper most threshold which is n minus 1 delta. So, in that case all the comparators will have output 1.

So, there is some kind of thermometer code that we are seeing as a signal goes on increasing the number of one goes on climbing. So, we can see that this is the mercury of the thermometer. So, one can be seeing as the mercury of the thermometer the high output coming from the comparators can be seen has the mercury level in our thermometer code as the signal is increasing the number of once is increasing or in another words the number of comparator going high is increasing. And of course this is not the, this is not the binary code this is the thermometer code and therefore, from here we need an encoder which can convert this into our binary codes 7 bit binary code. And for that we need to just look at the corresponding truth table and arrive at the digital logic.

So, we are going into detail we can just see that there is going to be and an encoder. An encoder which is basically taking in these n minus 1 inputs. So, you have out puts

starting from C 1 to C n minus 1. So, these are C n minus 1 inputs coming in and there is a digital encoder, encoder logic which is suppose to provide us our desired 7 bits you can call it d 1 to d 7 corresponding to this n minus 1 inputs. So, basically we can write to table for each of these outputs which of these 7 outputs in as a function of these n minus 1 inputs and then, simplify the logic function and arrive at the logic implementation of the individual outputs. That can be done using a k map and computer can solve it because you have such a large dimension of inputs.

So, CAD tools can solve this truth table and give as on optimizes Boolean logic, gate level logic implementation of this digital logic function. Where you have n minus 1 input and each of them can be 1 or 0 and you have corresponding in to that 7 outputs each of these 7 outputs are some logic functions of these n minus 1 inputs. Which can be upgraded using CAD tool or you need to do this specify a truth table where each of these output the 7 bit outputs will be specified as the function of these.

So, this is simplest possible logical implementation of an a to d converter, where we are using analog blocks namely the comparators and after that the digital outputs obtain or encoded with the help of a digital logic. And here we can see of course, that if you want higher resolution of the ADC; that means, you gone larger number of bit is in your ADC then of course, the number of comparators required will go on increasing exponentially for 7 bit it is 127 ADC s for 8 bit 255 sorry, for 7 bit it is 127 comparators for 8 bit 255 comparators and so on.

So, the number of comparators required will go on increasing strictly as the number of bits increase. And in general the flashed ADC topology is not implemented for more than 5 to 6 bits, beyond 6 bit is become very costly in terms of number of comparators that are required. So, if the resolution required it is less than say 6 bit then we can still go for flashed ADC and we in the one of the advantage of the flash ADC that it is very fast in one short it can give you the desired binary codes the input comes over here and immediately you have the entire 7 bit available at the output of an encoder in single cycle. So, it can be a very fast, but of course, you have larger number of comparators implementing it and exponentially it goes on as the number of comparators and hence the area goes on increasing exponentially.

So, we can have relatively simpler versions alternate versions where we can mitigate this disadvantage and use fewer number of comparators to implement our overall functionality. And we are going to a looking to one such topology which is going to be highly suitable for the signal processing we are trying to do and specifically the kind of signals characteristic that we have. Of course, choice of ADC is very much dependent upon the signal characteristics. If your signal is having relatively high frequency contained then of course, you will try to choose ADCs which can process a signal faster and use the digital output faster. If the signal content the frequency connected to the signal is relatively low I do not worry So much about the sampling speed or the conversion rate of the ADC.

In some applications you may required very high resolution of ADC. We are all aware of the standards of imaging for example, audio where people may go for 16 bit 20 4 bit is of data and there of course, we may not be at all choosing simpler topology is like the flash ADC. So, there the more important concern would be the resolution and there we may need to go for ADC which require much higher, which can give us much higher the resolution. And again there are some applications where both resolution as well as speed becomes important and that kind of ADC is become more and more challenging to design.

So, here in our case we are looking at the biometrical signal which is relatively low frequency signal. And also an application which is having moderate resolution, we may be able to do the required processing with just 7 bit of resolution that is moderate or relatively low resolution. You can have application we have just 4 bit may be sufficient, in this application looking at neural signals accusation and processing. People have shown that 7 to 8 bit of precession may be good enough for extracting the required information in the digital domain.

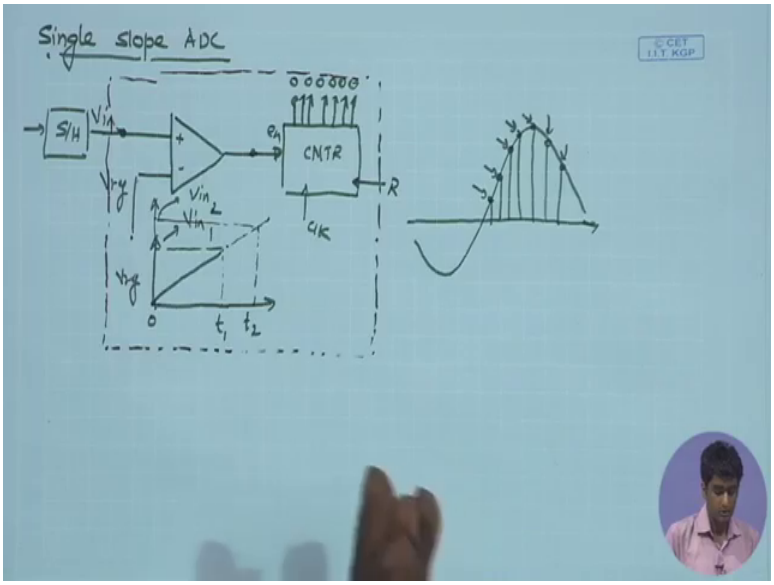
So, the target of the ADC resolution comes from the may be data processing requirements. So, if you are running some algorithm or trying to implement some algorithm in the digital domain that will tell you how much bit of precession ADC must have, So that you know correspondingly how much resolution how much accuracy the analog domain processing much preserve so that the required information can be extracted in the digital domain. So that information coming from some established studies which are telling us that 7 bit of precession is good enough for the data that we

are targeting. And then we also know the kind of frequency content that we have in our data.

So, starting from this high level requirement the data rate and the corresponding resolution we are going to arrive at the ADC topology. Were first fall going to choose and ADC topology which is suitable for this particular signal and then we are going to looking to the block level implementation of that ADC trying to arrive at the specifications of those ADC. Then go deeper step by step and deal with the intricacies of the block and the circuit level implementation.

So, let us let us look at an alternate scheme that we are going to focus on and studying deep that is call single slope ADC relatively simple architecture, but it is good enough case study through which we can show a lot of delicate details related to the circuit level implementation of the ADC.

(Refer Slide Time: 19:55)



So, the single slope ADC is just flip case of a flash ADC, where rather than having n number of comparators and So many different references we have just one reference which is being swept. So, in the single slope ADC we have only one comparator single slope ADC we have only one comparator and rather than comparing the input signal with. So, many different levels we compare it with the changing reference linearly increasing reference.

So, the overall scheme for such in ADC can be represented as block level. So, you have first of all the input signal coming say at the negative positive terminal of the comparator. And you have the V ref which is increasing linearly and you have the comparator output V, which is supposed to detect the crossing point when the V ref basically approaches or a crosses V in. So, V ref is supposed to start from 0 if your lower supply is 0 suppose start from 0 that is the minimum input level possible that the ADC can achieve or receive, and it is supposed to be increase linearly with time.

So, the V ref is supposed to go on increasing with constant slope ideally and we are interesting finding out the time at which V ref crosses the V in. So, if the V in is larger then of course, V ref will take longer time to reach V in if I say that this is your V in 2 and this is V in 1, if V in 2 is garter than V in 1. Then of course, we can see that if I assume that the slope of the V ref is constant that t 2 will be linear related to t 1 and as V in goes on increasing, we will have t 2 proportionally increasing exactly proportionally increasing with respect to V in. So, t 2 will have exact proportionality with V in and as result that t 2 or basically the time of comparator flipping captures the magnitude of the input signal. If the input signal is larger the time of the comparator flipping will also be proportionally larger exactly proportionally larger. So, it is basically capturing the magnitude of the input signal.

So, all we have to detect is that if the comparator has been say initialized with 0, initially when the V ref is 0 and V in is of course, higher than that minimum value delta V or higher. Than 0 then of course, according to that the comparator output will be high to begin with and then, as the V ref goes on increasing ramping ultimately at some time t the output over here will be going low, and that time t is the major of the digital value sorry, the analog value. And how do we measure time? One of the most convenient way is to use a counter sorry, you can have digital counter which is having some particular clock given to it and the enable signal is coming from the comparator.

So, you are having an enable signal which is basically the comparator output. And whenever the comparator output is falling low the counter stops counting. So, it has to be initialized with say a 0 count. So, there should be, there should be a reset input So that at the initial of phase of the comparison when the comparison is starting the comparator the counter output are all set to 0. So, in our case we are expecting that we are going to have 7 bit, counter because we need 7 bit word for the analog value.

Initially it should be starting with all 0 and the clock should continuously increment the counter output from all 0 towards the maximum value which is 1, 1, 1, all 1s. And the counter has to stop whenever the input signal reaches the different signal and that particular count produced by the counter at that time is basically going to quantize the total time taken by the ramping V ref to reach the input signal. Hence that digital word is going to be the quantized value corresponding to our input signal. So, that is the simple concept regarding single slope ADC, and then we can look into the overall implementation of this scheme.

Now so, we will be proceeding step by step and going in to the detail of each of these blocks. So, let us let us start with the very first block which is going to come before the comparator unit. So, that unit is called the sampling block. So, before the analog data can be digitized by our ADC which can be basically denoted by this entire black box and it is constituents, the data must be stable, it must be sampled and it must be at the value, because ultimately we are trying to digitize samples of the analog data.

So, if you have a continuous time analog data coming in. We need to first fall in the time domain sample the continuous time data at specific instances. And then obtain the digital value corresponding to these samples. And we know that the maximum sampling frequency determine by the neck straight. If we know the maximum frequency content of the signal we can determine the sampling rate has 2 times that frequency. And that is going to give me the interval between the sampling points. And these are the analog values which are sampled, which are of our interest and which are we which are going to be converted into the corresponding digital word representation.

So, we will be looking into the details of the sampling unit over here which although may look simple, but there are several design issues associated with the simplest block over here which is a sampling hold circuitry and we looking into the transistor level implementation it related non abilities design issues associated with that and so on.

So, let us stop here. Before we go in to the details of this sample and hold unit, I just stop here and see whether there is any question.