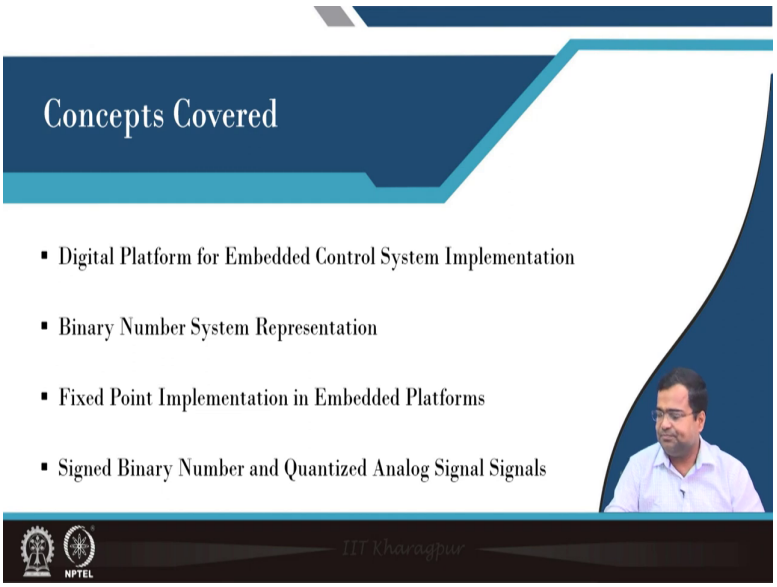
Digital Control in Switched Mode Power Converters and FPGA - based Prototyping

Prof. Santanu Kapat
Department of Electrical Engineering
Indian Institute of Technology, Kharagpur

Module - 07
Introduction to Verilog and Simulation Using Xilinx Webpack
Lecture - 67
Fixed Point Implementation in Embedded Control System

Welcome. In this lecture, we are going to talk about Fixed Point Implementation in Embedded Control Systems.

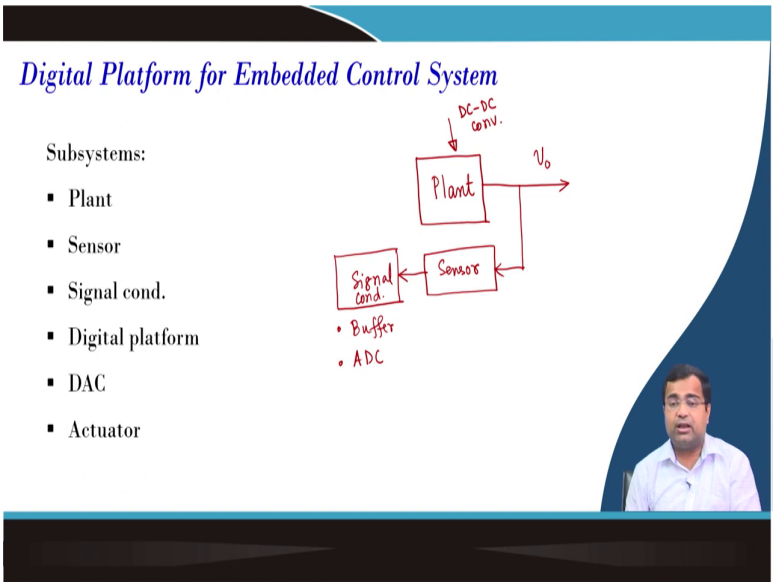
(Refer Slide Time: 00:30)



So, in this lecture, we are going to consider the digital platform for embedded control system implementation, then what is the binary representation for system representation because any digital platform only accepts binary data, either 0 or 1, digital data? So, we need to understand the binary system representation.

Then, the fixed point implementation aspect in the embedded platform. And finally, how to deal with significant numbers, sign binary numbers and how can we correlate such sign binary numbers with an actual real-world analog volt, an analog signal which is quantized.

(Refer Slide Time: 01:07)



So, in this case, we are talking about the digital platform for embedded control systems you know. So, if we consider any control system, whether it is a power converter or in typical consideration, so we will have one which is called a plant. So, this is like our plant and it can be, in this case, it can be simply our power converter.

Then, after the plant suppose we have some output, so we want to link that this output in this case is the output voltage. So, this can be a simple DC-DC converter, DC-DC converter. But it is not necessary this embedded control system can be applied in many branches of engineering. But here we are considering the DC-DC converter as a case study.

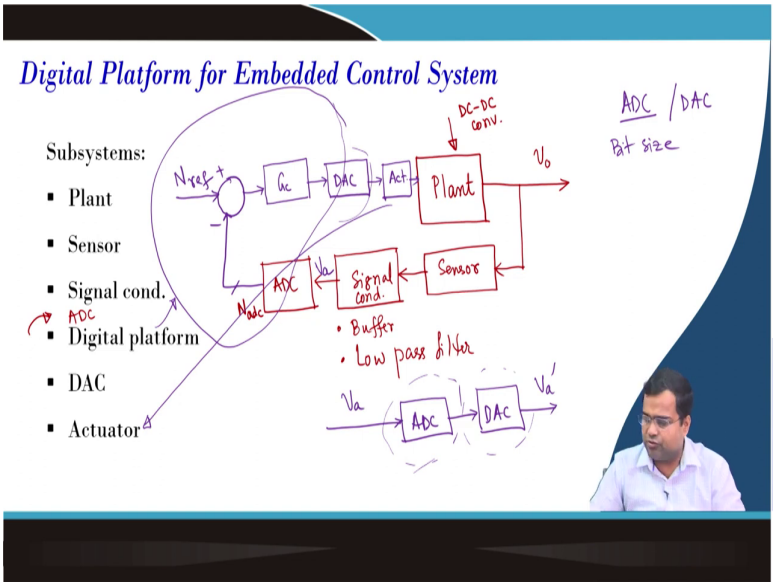
Then we need to sense this voltage. So, how do you sense? So, generally, there will be a sensor, ok, and this power converter, so let us say the sensor. If we are only talking about the voltage sensor then this sensor; means, voltage is very straight forward we can sense it.

But to convert it into the digital signal or to give this voltage to the ADC input, we need to go through a certain state; that means, we need to step down because it has to be mapped with the span of the ADC. So, this sensor in the case of the DC-DC converter typically has a resistive divider, so that you can step down the voltage. So, that you can comfortably take into the analog voltage span.

Secondly, you may have to put some impedance matching network, so that the ADC will not be loaded. So, that means, some amount of signal conditioning circuit is needed which can be

the next step; that means, some signal conditioning circuit. So, this signal conditioning circuit includes this circuit includes your buffer circuit for impedance matching and all. And it also includes your A to D converter, ADC; that means, we have an ADC circuit.

(Refer Slide Time: 03:32)



And so this is a part of signal conditioning circuit which is like ADC because we are talking about a signal conditioning about or even you know to be more precise, that means if you want to. So, let us say this is just a buffer with a low pass filter, it consists of f low pass filter because we need to reject some noise.

Let us say we are separately considering ADC; that means, this will be ADC. So, here we should list one ADC in the list. This act means, the input to the ADC in another g signal, so we need to make sure the signal conditioning output should match the span of the ADC.

Now, this ADC data will be digital. So, it can be a number, it can be ADC data. This number you know if we consider this number, this number goes to, ok; this number goes to our feedback signal. That means, this is a minus and here we have some positive reference comments.

Then what we will have? We will have a digital controller; that means, you know we can have a digital controller. For the output of the digital controller in some cases we may need some DAC and in some cases, we may need an actuator. So, it is an actuator, so which is this actuator?

Now, for a power converter, and actuator you can think like it is the gate drive circuit, that can be an actuator because you have to turn on and turn off the physical MOSFET switches. But you may not need DAC because you are generating the gate signal.

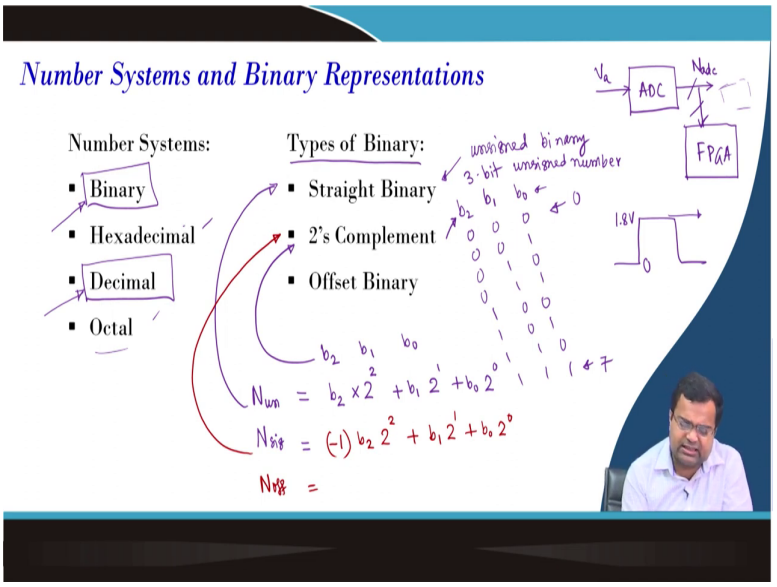
But suppose we will be talking about mixed signal current mode control where you need to consider the DAC also. So, we will go to that point. But what I am trying to say is that up to this part is in your digital platform, right? So, since it is a digital platform, this data which is going inside is basic. So, this data can be 8-bit depending upon whether we are talking about ADC.

So, we have to consider what is the bit size of the ADC, bit size. Is it 8-bit ADC? 9-bit ADC? 10-bit ADC? The same thing is also applicable for DAC, DAC bit size; that means, how in 12-bit DAC, and 10-bit DAC because the input to the DAC and output of the ADC are binary numbers. And so we need to understand, anything that we represent in the digital platform is a binary number.

So, we have to understand some basic philosophical aspects of the binary number system as well as how can we interpret such binary numbers into real-world signals because here the input to the ADC is an analog voltage, so how it is mapped to the analog voltage? Or suppose we have an analog voltage, then if we pass through an A to D converter followed by a D to A converter we will get V a dash.

So, it is a natural question whether V a dash will be exactly equal to V a or whether there will be some effect of quantization because this will have a finite word length of 8-bit. 9-bit, it cannot be an infinite bit. It will also have a finite word length. So, the actual analog signal has infinite resolution. But when it passes through ADC and DAC, it will have a finite resolution. So, naturally this V a dash will have some difference with V a in terms of quantization limit. So, that also we are going to consider.

(Refer Slide Time: 07:07)



So, here in the binary system when you are talking about the number that we are accept accepting as the input to the ADC sorry, output of the ADC; that means, we are talking about an analog-to-digital converter, we are going to see soon and output of the ADC is a binary number.

Now, this binary number is typically what comes out from the ADC, they are a line. That means, if we have an A to D converter, it depends, on your analog voltage there this is the ADC. So, whenever you select ADC you know you will have a choice of whether the data interface is serial or parallel. So, if you take a serial ADC, then it is just one number; that means, the ADC is serial. So, naturally, if the 8-bit ADC, is an 8-bit ADC. So, 8-bit will go serially.

So, first, it will start with a start bit, then it will have MSB, and LSB depending upon the type. So, it can be a successive approximation, whatever ADC will use, but in the case of serial ADC, we need to understand the data format; which data comes from it is the MSB or LSB.

Then, similarly, one by one data will come and then once the data is over it will just start the conversion. So, we should not mix the data of a particular corresponding a particular sample with the data of the next sample. So, you should not mix up. You should separate it.

But as is as you know that serial ADC has only one output wire or connection, so naturally the data will come serially. So, the conversion process takes time. But you can also have a parallel ADC where the number of lines can be parallel. So, you will get the data parallelly; that means, it can be 8-bit, or 10-bit, but here the number of data lines will increase.

So, as a result, when you talk about a digital platform you need more pins of the digital circuit to accept that parallel data from the ADC, ok. There is another possibility because sometimes we can use a parallel ADC which can be a high-speed ADC. But suppose you have limitations in terms of the IO pin of the digital device then what you can do; I am talking about let us say this ADC data is going to a let us say an FGPA that we are going to discuss. So, this data is coming.

But suppose the FGPA does not have sufficient IO pin because you are constrained in terms of cost, in terms of area. So, then what can you do? Your data output from the ADC is serial parallel ADC because we want a fast ADC, typically serial ADCs are slow. But you do not have a sufficient pin.

So, you can use some kind of interfacing digital circuit which converts parallel to serial and that will be a high-speed circuit because it does not need any conversion. It is just a codification. That means, you are converting parallel data into serial and it is a coding encoder and decoder that have to link, and in that way you can also manage.

And this circuit can be a very high speedy encoder and decoder circuit, that can operate at a very very high frequency. So, you can through there will be a delay, of course, still there will be a much smaller than actual serial ADC. So, in summary, whatever ADC you take, whether it is serial or parallel. these lines are a number and depend upon the logic level because now we are talking CMOS logic, which can be 1.8 volt logic, earlier it is to be 3.3, 2.5. So, it will only generate either high or low.

When it is high if the logic level is 1.8 volt, it will be 1.8 volt or it will be 0 volt. So, each wire will carry the voltage which can be can take a value from 0 or 1 which corresponds to a certain voltage. So, that means, it is a binary number. But once it comes inside, then we can represent it in different number systems either binary, hexadecimal, decimal or octal.

Generally, in hexadecimal you can take 4 binary together, it is very easy from 0 to 16. Then octal 0 to 8, actually 0 to 7 can create a 3-bit number. So, these are the representation of a

binary system where the data comes in inside a digital platform. Now, we are talking about a digital platform inside.

So, we have n number of choices in terms of data. But in this lecture, we are primarily dealing with this binary number and the decimal number because decimals are very because we all deal with decimal numbers 0 to 1; we are very familiar with the decimal number. So, sometimes we will use this decimal number to give you some kind of thought in the code, but we have to specify by default Verilog takes a decimal number. But if you want to specify binaries we have to mention that is a binary.

Now, out of this binary number that we are dealing with, since we will be dealing extensively with binary numbers, the binary number can take a different form. It can be straight binary which is unsigned binary. You can say unsigned binary that is typically we deal with unsigned binary which is let us say you know if you take let lookup table; that means, you are talking about b 1, b 0, and b 2.

So, in the look-up table, what we will take? We will take 0 0 0, 0 0 1, 0 1 0, 0 1 1, 1 0 0, 1 0 1, 1 1 0, and 1 1 1. So, these are the 3 possibilities and this is a straight binary, where it represents a number between 0 to 7. So, this is 7, this is 0. So, it is a 3-bit number. Now, in this representation, there is no sign concept.

So, it is just a number that varies between 0 to 7. But suppose the same 3-bit; that means, it is a 3-bit unsigned number sorry; 3-bit unsigned number which is represented by b 2, b 1, b 0. This is the MSB and this is the LSB that you know. Now, we want; that means, b 2, b 1, b 0. Now, we want to represent a significant number using 3-bit. How does that go to?

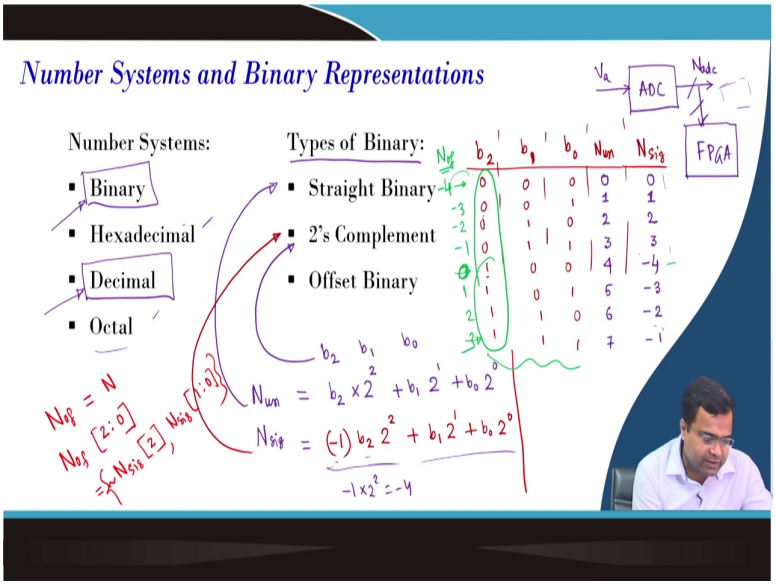
One of the ways is a 2's complement which deals with sign numbers. So, how do we deal with 2's complement? So, let me represent this b 2, b 0; that means, you have a bit that can take either 0 or 1, and each of these bits can be either 0 or 1. So, what will be the number in the unsigned number? That will be b 2 into 2 to the power, 2 to the power 2 plus b 1 into 2 to the power 1 plus b 0 into 2 to the power 0. So, this is an unsigned number that goes back to this, straight binary.

The same number if I can use a sign, then how does it look like in 2's complement? So, I am representing in 2's complement. So, this will be minus 1, then b 2 everything else will remain

the same, b 1 2 to the power 1 plus b 0 2 to the power 0. That means, in 2's complements; sp. this goes back to our 2's complement.

That means, if, ok; another is the offset binary. So, let us say we can have offset binary. Again, this is a representation where I will talk to you, so I will tell you it is just you have to complement the MSB; that means, we will take it later. So, let us say in unsigned number if all bits are 0; that means, you know if we just rub up this particular table. So, let us say we have a number, ok.

(Refer Slide Time: 15:47)



So, let us say we have a truth table which is b 2, b 0, b sorry b 1, b 0, ok and we want to give the unsigned number and signed number (Refer Time: 16:05). So, 0 0 0, 0 0 1, 0 1 0, 0 1 1, 1 0 0, 1 0 1, 1 1 0 and 1 1 1. So, in the case of an unsigned number, this is 0, in the case of the sign if you replace here all the bit 0, again it will be 0. So, both are the same.

When it is 1, the unsigned number is 1. And you can see 2 leading bits are 0, so this is also 1, there is no problem. 1 0 again it will be 2, it will be 2. 0 1 1 it will be 3, it will also be 3. But the difference comes when the MSB changes. When the MSB becomes 1, in the case of an unsigned number it is 4. But, what will be the sign number? 1 all 0. So, this bit will be as if minus 1 into 2 to the power 2, so it will be minus 4. So, this will be minus 4. Then, what is next? This will be 5, this will be minus 3.

Then this will be 6, this will be minus 2. This will be 7, this will be minus 1. So, that means, here you can see the sign number can vary between minus 1 to 3. That means, what is the range of the significant number? The lowest is sorry, the lowest is minus 4, highest is 3. In the case of an unsigned number, it varies between 0 to 7.

So, what we will find is that effectively the sign number in 2's complement you can take a bipolar digit; that means, from minus to plus, whereas sign. Because in the A to D converter that we are dealing with it will be a sign because we have to take the error voltage which will have a sign. So, that is why understanding this is important.

Now, going to the offset. So, if you take an offset binary number, so the offset binary. In offset binary, in this case, it is a 3-bit number. So, this will be simply whatever is in 2's complement, right? So, you have to take the first; that means, I am just taking the offset number as 3-bit, which is 2 to 0, 3-bit.

This will be equal to you can say the 2's complement which is a significant number of the complement of this MSB; that means, you are simply inverting. That means, whatever you will get this bit, whatever number we will get will be simply inverted. And then the rest of the bit. So, we will talk about concatenation.

It will be the sign number 1 to 0; that means, in 2's complement, so that means, if we take the sign number, so sign number starts from 0 to all 1. That means, in the case of offset binary, what will happen? The offset binary because in compared to the 2's complement since you are inverting this particular data, right? So, we are inverting this data. So, we are inverting this data, the whole thing, this whole thing for 2's complement.

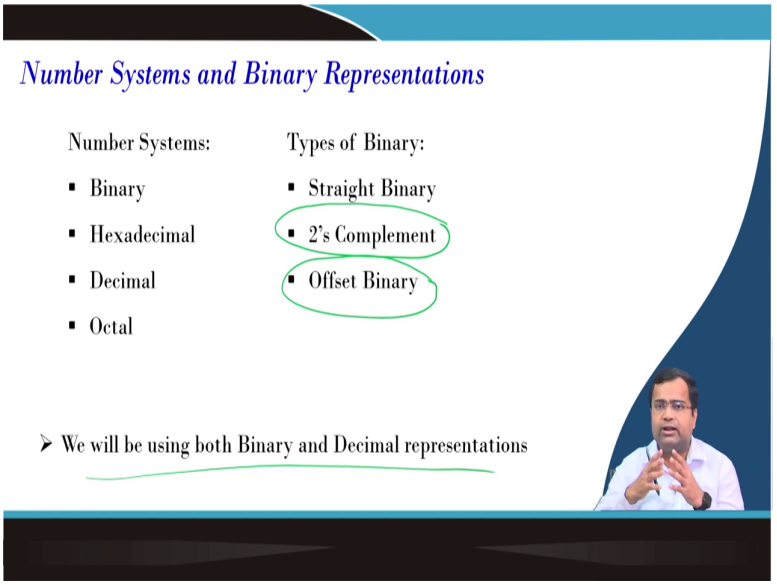
So, in offset binary all; that means, 1 all 0 will be 0. That means, in the case of offset binary what will be the number? All 0, that means, it will be 1 all 0. That means it will be just 1 all 0. So, that means, this number 1 all 0 will be the 0. And all 1 will be positive and all 0 will be negative.

Because in offset binary 1 all 0 in 2's complement, it will be the maximum negative, ok. And all 1; that means, in offset binary if you represent this 0 to 1, it will from maximum to minimum. So, you can represent a negative bipolar number by suitably making the offset. That is why it is called offset binary.

So, that means, the offset binary case of offset binary will be the maximum negative number. That means it starts with a maximum negative number, which will be minus 4, then minus 3 because if you complement, in 2's complement 1 all 0; that means, 1 all 0 corresponds to minus 4, ok. Then you will get the next digit 1 0 1, 1 0 1 will be minus 3.

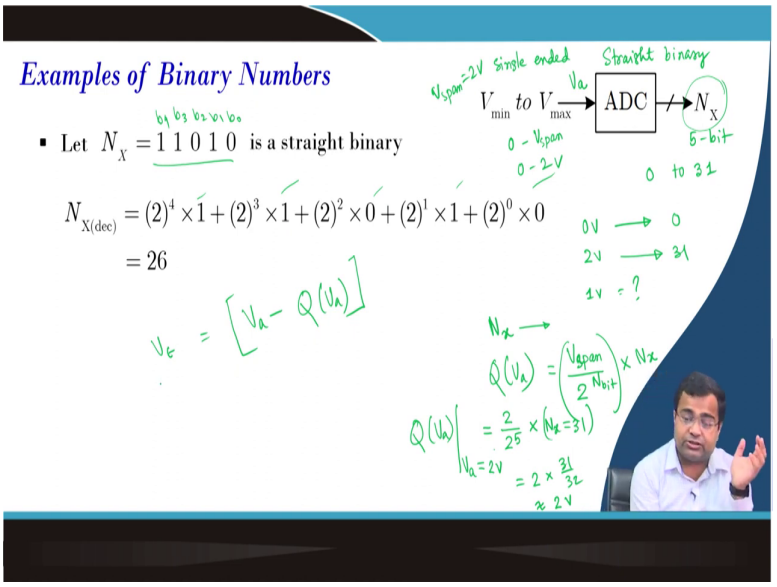
Then, next will be 1 1 0, and 1 1 0 will be minus 2. Then it will be 1 1 1, 1 1 1 will be minus 1. Then, the next one, your compliment is 0 0 0, it should be 0, then 1, then 2, then 3. So, that means, in offset binary, all 0 will be minus 4 in this case and all 1 will be 3. So, it will vary from minus 4 to 3. And this is typically needed in our case because you are using a D-to converter which has to be converted into offset binary. So, we need this thing, ok.

(Refer Slide Time: 21:39)



So, that means, in our case we need you to know we need both this number decimal will be used in this representation. For ADC, we will be using 2's complement. For DAC, whatever we are sending to the DAC digital data should be in the offset binary. Because this comes from the data sheet of the manufacturer and here we are using analog devices ADC and DAC. So, we are making that data format according to their specification.

(Refer Slide Time: 22:04)



So, an example of a binary number; is that means, N x is a straight binary. So, you can represent. So, that means, if you represent what we have discussed it is a 5-bit number. So, the first it will be b 4, then b 3, b 2, b 1, b 0. So, b 4 1, b 3 1, b 2 0, b 1 1, b 2 0. So, the number will be 26. What will be the maximum number? All 1. So, that means, 2 to power 4 is 15. Then 2 to the power 5 will be you know it will be 32, so 31, ok. So, the maximum number will be 31.

Now, the next question. So, we have data which is let us say 5-bit ADC, now this 5-bit ADC is in the form of state binary because as per the data, see let us assume that this ADC straight binary. That means, his number can vary between 0; that means, it can vary between 0 to 31. But we need to link how it is linked with the analog voltage because it is dealing with a voltage.

Let us say it is single-ended data. What is single-ended data? That means, it is positive data. So, it can vary let us say between 0 to b span, in this case, let us say the span is 2 volt. So, it varies between 0 to 2 volt; which means 0 volt is mapped to the number 0 and 2 volt is mapped to the number 31.

So, what will be the voltage for 1 volt? What will be the number? So, we need to map this data. That means, we need a scaling factor. So, how do we scale this number? So, whatever N x we are getting, if we want to convert into this because now we are talking about a voltage

equivalent voltage, but you remember in analog whatever we are applying to the ADC it is un quantized; that means, it had it has an infinite resolution.

But suppose you think that I want to represent this data into an equivalent voltage, then there will be a quantized voltage. So, I would say that; if this is your V a, then quantized V a. What will be that? So, it depends on the span of the ADC; this is the that means, the span divided by; divided by what? The 2 to the power bit is the number of ADC bit. This whole factors into your N x in this case. That means if N x is 0 you see the quantized voltage is 0 because the analog voltage is 0.

But if the analog voltage is 2 maximum volt, then we are getting N x to be 31 which is the upper limit. Then, what will be the quantized voltage? So, that means, the quantized voltage for V an equal to 2 volt this is analog voltage will be equal to; what is the span? It is 2 volt divided by 2 to the power 5. 2 to the power 5 means 32 into the number will be the maximum. So, N x will be, N x will be equal to 31, ok.

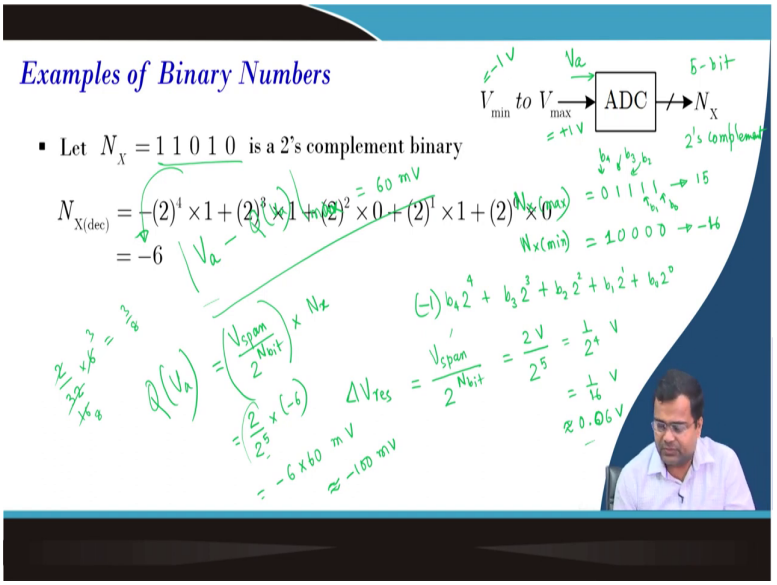
So, what we will get? You will get 2 into 31 by 32 which is approximately equal to 2 volt, but it is less than 2 volt. It is not exactly. So, that means, the quantized voltage level is different from the actual voltage level. And this difference will; mean, the difference between the quantized voltage level minus or the actual voltage level, I would say, this difference will that will say the error voltage.

This particular voltage; this error can be large if the bit size is small because the resolution will be poor. If the bit size increases then this error can be reduced. So, the bottom line is this we have to be very careful about this quantization level because if we use a very lower bit ADC, then the ADC cost will be smaller. The speed of the ADC can be also faster because this bit has to be processed for the digital processor. So, the processor width will also be smaller.

Because if you deal with the 5-bit data, then all the controllers and all the filtering operations will be dealing with the 5-bit input data. But imagine if it is 10-bit data, then you have to do all the arithmetic operations with 10-bits. So, naturally, the size of the hardware architecture will increase, power consumption will increase, and the propagation delay increase, but the resolution will decrease, and resolution will improve.

So; that means, it is a trade-off between the accuracy and the power consumption or your computational time. So, you have to be very careful about the selection of ADC because too large a bit size can cause your hardware architecture cost too much and that may not be acceptable, ok. So, I think we understood the state binary.

(Refer Slide Time: 27:58)



That means this is the case example. So, this is in 2's complement. Now, suppose this signal; again, this is an analog signal which varies between V min and V max. Let us say V min is minus 1 and V max is plus 1, 1 volt, and again let us say it is a 5-bit number. So, now this 5-bit in 2's complement. So, the natural question is, what is the N x max in 2's complement? It will be 0 all 1. What will be the N x min? Because we are talking about 2's complement it will be 1 all 0, ok.

And what will be that? This corresponds to what? This corresponds to 15. And what will be this in 2's complement? This corresponds to minus 60. Why? Because we know minus 1 because this position is my b 4, this position is b 3, b 2, b 1, b 0. So, this will be b 4 into 2 to the power 4 plus b 3 into 2 to the power 3 plus b 2 into 2 to the power 2 plus b 1 2 plus like 1, b 0 2 to the power 0.

So, if all bits are the leading bit is 1 all 0 that is the maximum negative or is the minimum value. So, that means, we can represent minus 16 numbers corresponding to minus 1 volt and 15 numbers. Now, you can ask me what is the resolution. So, the resolution; that means, if I

say delta V resolution, again it is the span of the ADC divided by 2 to the power of the number of bits.

So, in this case, the span is varying between minus 1 volt to plus 1 volt. So, it is a 2 volts span. And what is the bit size? It is 2 to power 5. Then what is the resolution? So, the resolution will be 1 by 2 to the power 4 volt which is 1 by 16; that means, you know 1 by 16, yeah. So, 16 volt. And which can be roughly around 0.06 06 volt approximately or something 0.06125 like that.

So, what I am trying to mean, is this is a resolution, so there will be an error with the analog voltage; that means, the quantized voltage level minus your quantized voltage level, this difference if I take the minimum value of this difference sorry the maximum difference will be in the order of 60 millivolt. Minimum, if both matches let us say if you take 0 volt, the output is 0. So, then perfectly there is no error.

So, typically the error can vary between plus and minus. So, I am talking about the maximum magnitude of the error can be 60 millivolt. And sometimes this may not be acceptable because if you talk about a power converter with 1 volt output then because of this quantization you may lose regulation up to 60 millivolt which may not be acceptable.

So, in that case, we have to increase the ADC bit size and that will define us; that means, the ADC bit is defined by what is the resolution. But again this 60 millivolt is the raw data. Suppose. we have a 1 volt output then there is a step down like a resistive divider. So, it will further decrease, and the resolution will further degrade. And we will discuss this aspect in the subsequent lecture.

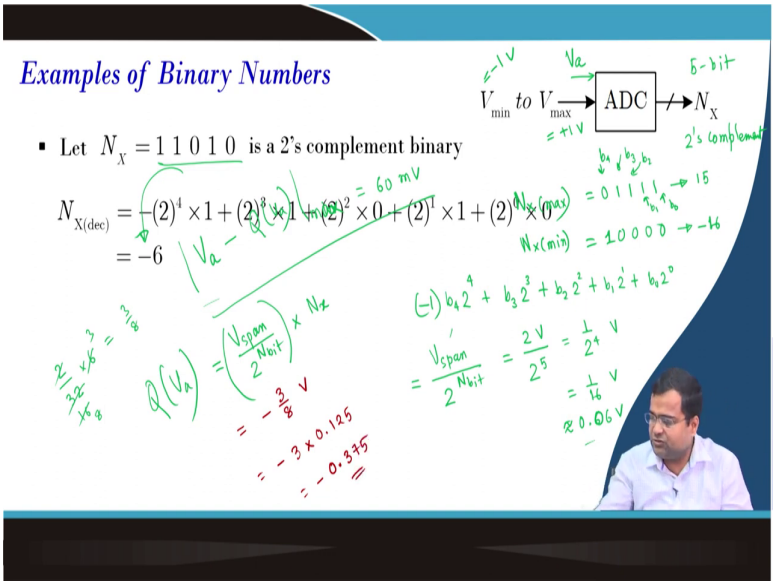
But the bottom line, we need to represent this binary number in 2's complement into a real-world number and that is why you know how can we link with this. That means, how can we realize this real-world number? Again, in this case, will be your V span by 2 to the power N bit, this factor multiplied by N x and N x can be minus to plus and that is what we have represented; if this particular case it gives rise to minus 6 volt. So, you can realize what will be the analog voltage for this case.

So, what is the analog voltage in this case? It will be 2 by 2 to the power 5 into minus 6. And we know that this ratio is 60 millivolt, so that means, minus 6 into 60 that is a millivolt. So, it

corresponds to roughly point 100 millivolt, yeah, it is roughly equal to 100 millivolt; no. So, 1 volt; roughly equal to; is it? 60 millivolt is the resolution because it is, ok.

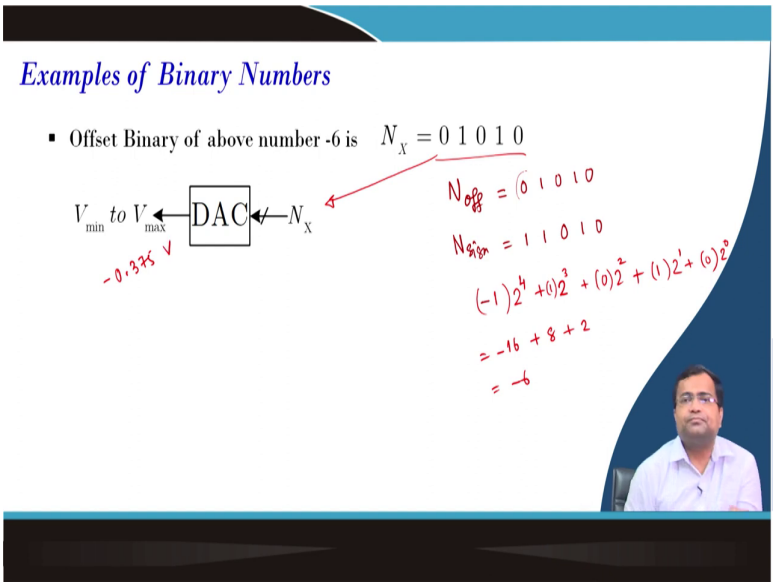
Let me write down again 2 to the power 5 is 16. So, it is 2 by 32 into 6 which is you know 2 16, then 2 8, 3; so, 3 by 8, 3 by 8 sorry. It should be which means; yeah, so which means for this particular case it will become minus 3 by 8 volt.

(Refer Slide Time: 34:16)



So, what will be that? Minus 3 into, so 0.125. That means, minus; yeah I think we are right. So, in it should be; there was a calculation mistake. It should be 0.3 you know 75 so volt, ok.

(Refer Slide Time: 34:50)

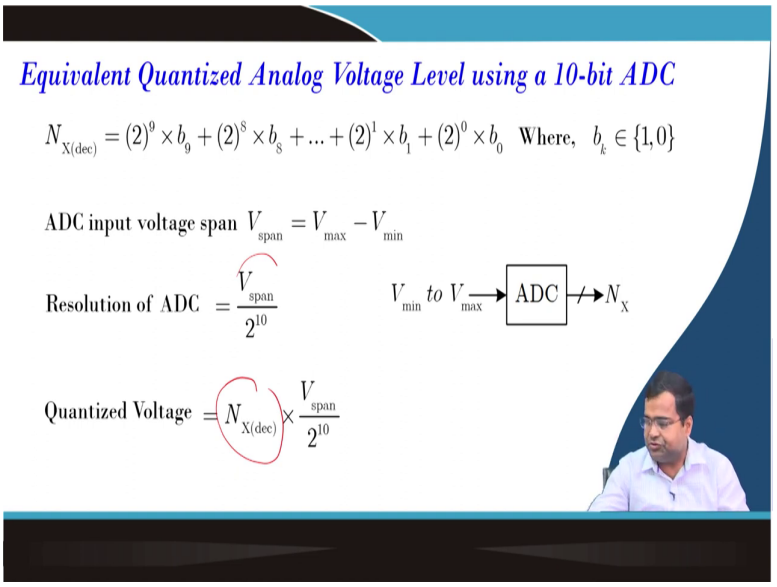


Now, binary number. So, if it is an offset binary A to D converter, then what will be the offset binary? Because offset binary means it is the I would say the complement of the MSB of the; so, that means, if this is there say we are talking about offset binary which is represented by 0 1 0 1 0. So, we need to understand what is the sign value of this.

This will be, this bits has to be complemented. So, 1 1 0 1 0 which means minus 2; sorry minus 1 into 2 to the power 5, then plus 1 into 2 to the power 4 plus 0 into 2 to the power 3 plus 1 into 2 to the power, it is a 5-bit number sorry. So, it should be, this will be 2 to the power 4, 2 to the power 3, 2 to the power 2, 2 to the power 2, 2 to the power 1 plus 0 into 2 to the power 0.

So, this will be minus 2 to the power 4 means 16, plus 8 then plus 2 which is minus 6. that means, the same representation which we represented last time, in offset binary it will be minus 6, which corresponds to the voltage of minus 375 millivolt. If this; so, this is a DAC, that means, if you pass a number in the offset binary form it will result in minus 375 volt, ok.

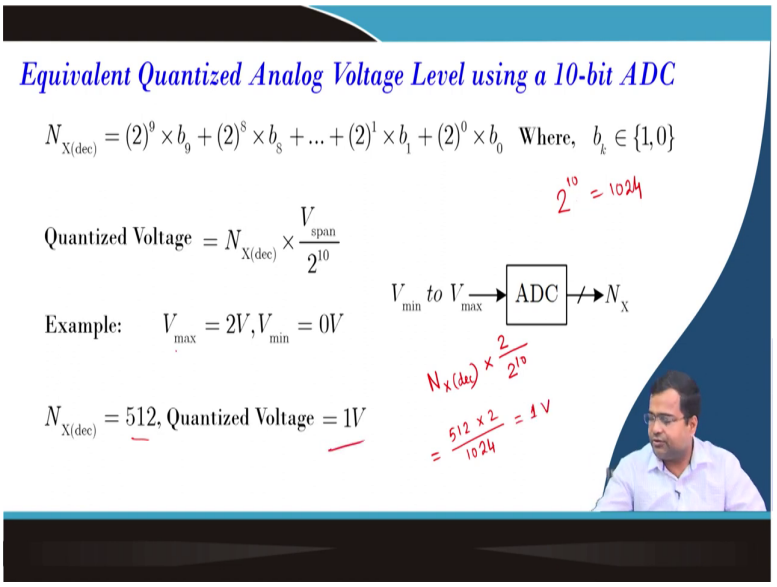
(Refer Slide Time: 36:54)



So, that means, if we talk about this equivalent analog voltage; so, we have discussed if it is a 10-bit number we can accordingly generate you know write this write-down. So, first of all, we have to understand what is the span that we have discussed, which is the range of the voltage swing which is compatible with that ADC.

Then, the resolution of the ADC we have discussed, a span of the ADC divided by 2 to the power number of bit. So, in this case, is a 10-bit ADC. Then, the quantized voltage level whatever number we are getting that multiplied by this factor, that we have discussed. So, it will be quantized. And we have seen there is an error between the quantized voltage and the actual analog voltage because of the finite resolution.

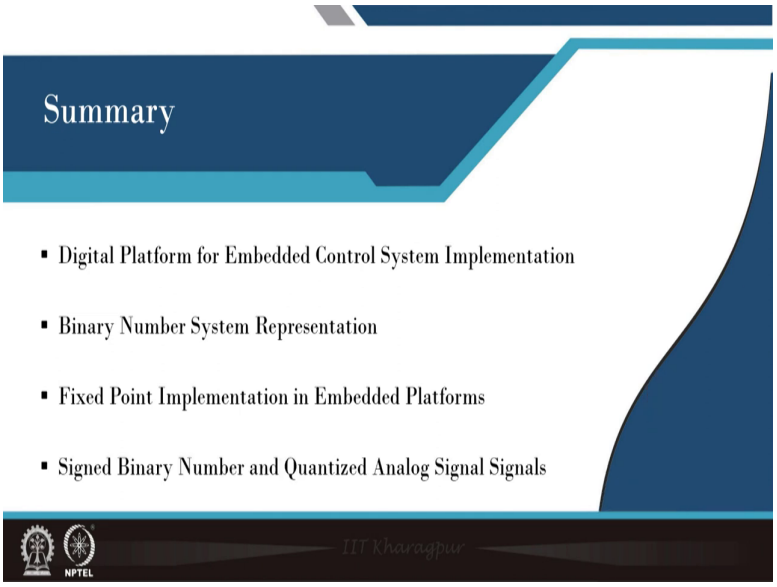
(Refer Slide Time: 37:37)



So, then with this quantized voltage; for example, if we take the maximum voltage on ADC as 2 volt, minimum 0. So, it is a single-ended. only single unipolar input voltage. So, what will be the quantized voltage level for a quantized if 512 is a number suppose? It is a 10-bit ADC. So, 10-bit ADC means, that means, you know. So, quantized voltage is what? For 512; 2 to the power 10 is how much? 1024.

Now, if you take N x into 512, then what we will get? First of all the resolution; which means, N x in decimal is 512. Into, what is the span? 2 divided by 2 to the power 10. So, it will be 512 into 2 divided by 1024, so which is equal to 1 volt. So, that means, the quantized voltage is 1 volt.

(Refer Slide Time: 38:43)



So, in summary, we have discussed digital platforms, some understanding of embedded control system implementation, number system of the binary. We have also discussed the fixed point implementation. And then, we also represent the sign number and how it can be linked with the quantized analog voltage if there is an A-to-D converter. So, that we can link with the real-world voltage.

Thank you very much.