

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 - 68
Fixed Point Arithmetic and Concept of Q Format

Welcome. In this lecture, we are going to talk about Fixed Point Arithmetic and the Concept of the Q Format. This lecture is the continuation of the previous lecture.

(Refer Slide Time: 00:34)

Concepts Covered

- Mapping Binary Number with Actual Voltage in an ADC
- Concept of Q Format
- Addition, Subtraction and Multiplication Rules in Q Format
- Examples of Arithmetic Operations in Q Format

IIT Kharagpur
NPTEL

In this lecture, we want to again consider the binary number system and we try to link how this binary number system can be linked with a real voltage when we are dealing with a 2D controller that is a gateway that represents a binary number and the analog voltage.

Then, what is the concept, how can we generalize such number system representation and equivalence between the analog voltage, and how can we link with a generic form so that you know we have to because, in the FPGA implementation, we will deal with a different type of coefficient of the controller. So, how can we make a generic representation that is called Qs Q format?

Then what are the rules for addition, subtraction, and multiplication in Q format finally, we will take an example of an arithmetic operation in Q format.

(Refer Slide Time: 01:24)

Introduction to Fixed Point Arithmetic

- Consider a binary $2+2$ bit number
4 bit number

$$N_x(\text{dec}) = b_3 2^3 + b_2 2^2 + b_1 2^1 + b_0 2^0$$

$$N_y(\text{dec}) = [b_3 2^1 + b_2 2^0 + b_1 2^{-1} + b_0 2^{-2}] \times N_{\text{scale}}$$

$b_3 \ b_2 \ b_1 \ b_0$

$N_x(\text{dec})$ varies from 0 to 15

video inset of a speaker

So, if I go to the fixed point arithmetic let us consider a 2 plus 2-bit number, basically it is a 4-bit number. So, that means, in the 4-bit number I can write something like this; means, I can write b_3 into 2 to the power 3 plus b_2 2 to the power 2 plus b_1 2 to the power 1 plus b_0 2 to the power 0 . This is one way of representation.

Now, I can also write in this way; that means, this b_3 to b_0 is the; that means, we have a number which is b_3, b_2, b_1, b_0 . And, in this way, if we write a straight; that means, you know N_x in decimal what we will get? So, it can vary N_x decimal, it can vary from varies from 0 to what? What is the maximum number? This is like 84 ; that means, it is 31 sorry, 15 because it is a 2 to 4 . So, 0 to 15 that is varies between 0 to 15 . So, if all are 0 bitst, then it is 0 . If all 1 it is 15 .

But, we can also write another representation let us say. Suppose we write b_3 into 2 to the power 1 plus b_2 2 to the power 0 plus b_1 2 to the power minus 1 plus b_0 2 to the power minus 2 . Now, how is it coming? You can think of the same number I just represented with a scaling factor.


(Refer Slide Time: 03:50)

Introduction to Fixed Point Arithmetic

- Consider a binary $2+2$ bit number $b_3 \ b_2 \ b_1 \ b_0$
4 bit number $N_x(\text{dec})$ varies from 0 to 15

$$N_x(\text{dec}) = b_3 2^3 + b_2 2^2 + b_1 2^1 + b_0 2^0$$

$$N_x(\text{dec}) = [b_3 2^1 + b_2 2^0 + b_1 2^{-1} + b_0 2^{-2}] \times N_{\text{scale}}$$

$$N_{\text{scale}} = 2^2$$


So, let us say we are talking about a scaling factor. What is the scaling factor? That means I can continue the same number here because we want to write the same number N_x in decimal the same number with a scaling factor. What is the scaling factor? So, here N_{scale} is equal to 2 to the power 2 because if you multiply 2 by power 2 these two are the same everything same. So, I took 2 to power 2 out. Now, why it is needed I am coming to that point.

(Refer Slide Time: 04:21)

Introduction to Fixed Point Arithmetic


- Consider a binary $2+2$ bit number $b_3 \ b_2 \ b_1 \ b_0$
4 bit number $N_x(\text{dec})$ varies from 0 to 15

$$N_x(\text{dec}) = b_3 2^3 + b_2 2^2 + b_1 2^1 + b_0 2^0$$

$$N_x(\text{dec}) = [b_3 2^1 + b_2 2^0 + b_1 2^{-1} + b_0 2^{-2}] \times N_{\text{scale}}$$

$$N_x(\text{dec}) = N_{\text{scale}} \times \frac{N_y(\text{dec})}{d}$$

$N_y(\text{dec}) = b_3 2^1 + b_2 2^0 + b_1 2^{-1} + b_0 2^{-2}$



So, I would say this to be this number I can say like an N_y in decimal. So, N_y in decimal is a scaling factor multiplied by N_x . Sorry, I am writing the other way around. So, it should be N_x , it should be N_y and this scaling factor can be different. So, the beautiful thing here is if I write in one case b_3, b_2, b_1, b_0 it looks all like an integer.

In the other case you see from here from this point or not, it looked like a fraction; that means, in the other case b_3, b_2, b_1, b_0 . So, that means, in the first case from this point or not decimal starts, but there is no decimal. But, here we have inserted a decimal concept; that means, when you write about decimal numbers in the binary sorry in fractions it will be like a $b_3 b_2 \text{ dot } b_1 b_0$, it is something like this. It is a dot; that means, it is a fraction position here.

But, the fixed point will never see a fraction it is just a 4-digit number, but you can either represent it in this N_x form or N_y form, it is a notional concept. But, the question is why do we need so?

(Refer Slide Time: 06:17)

Introduction to Fixed Point Arithmetic

- How do you represent quantized voltage for a 4 bit ADC?

assumption *unsigned*

V_{\min} to V_{\max} \rightarrow ADC $\rightarrow N_x$

0V 2V

straight binary

$$N_x = b_3 2^3 + b_2 2^2 + b_1 2^1 + b_0 2^0$$

$$N_y = N_x \times \frac{2}{2^4} = N_x \times \frac{1}{2^3}$$

$$= b_3 2^0 + b_2 2^{-1} + b_1 2^{-2} + b_0 2^{-3}$$

$$Q(V_a) = \frac{V_{\text{span}}}{2^{N_{\text{bit}}}} \times N_x = N_y$$

So, let us go to that number and let us consider this ADC. Suppose, this ADC let us say we are talking about the first unsigned number. So, let us say it is an unsigned ADC or I would say it is in the format of a straight binary assumption. So, this is an assumption, ok. What is the minimum value? Let us say it varies between 0 volts and it varies between 2 volts; that means, we are dealing with an analog signal where this is the analog signal is an analog input and we are getting a corresponding output.

Now, I want to write this data in such that I will get the quantized. So, last time what we did do? We did that quantized V_a to be what? The quantized V we got is a V span divided by 2 to the power N bit multiplied by N_x that we have written. So, here this whole thing we want to write as if in terms of N_y in decimal, right? So, this is in decimal N_y ; that means, how it is possible?

So, let us say whatever our N_x our N_x is it is a 4-bit number. This 4-bit number varies between $b_3 b_2 b_1 b_0$. And, then what is N_x ? So, N_x is basically what is N_x . N_x is b_3 into 2 to the power 3 plus b_2 into 2 to the power 2 plus b_1 2 to the power 1 plus b_0 2 to the power 0. Now, what is N_y ? N_y will be N_x multiplied by what is my V span it is 2 volt dividvoltsy 2 to the power what is the bit size? 4; that means, it will be N_x into 1 by 2 to the power 3.

Then, how do you represent? So, I can represent this like b_3 into 2 to the power 1 plus b_2 into 2 to the power 0 right 2 to the power 3, ok. Then b_1 2 to the power minus 1, it is divided by 2 to the power 3.

(Refer Slide Time: 09:30)

Introduction to Fixed Point Arithmetic *assumption unsigned*

▪ How do you represent quantized voltage for a 4 bit ADC?

$V_a = 2V$
 $Q(V_a) = 1.875V$

V_{min} to V_{max} → V_a → ADC → N_x → straight binary

$N_x = b_3 2^3 + b_2 2^2 + b_1 2^1 + b_0 2^0$

$N_y = \frac{V_{span}}{2^{N_{bit}}} \times N_x$

$N_y = Q(V_a)$

$N_y = b_3 2^0 + b_2 2^{-1} + b_1 2^{-2} + b_0 2^{-3}$

$N_y = 1 \cdot 2^0 + 0 \cdot 2^{-1} + 0 \cdot 2^{-2} + 0 \cdot 2^{-3}$
 $= 1 + 0 + 0 + 0$
 $= 1.000$

$N_y = 1 \cdot 2^0 + 0 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3}$
 $= 1 + 0 + 0 + 0.125$
 $= 1.125V$

1001
 $2^0 + 2^{-3}$
 $= 1 + 0.125$
 $= 1.125V$

So, it should be 0 sorry, minus 2 plus b_0 2 to the power minus 3. So, this is my N_y . What is the difference between this number and that number? What is the difference between this? What is the difference? What is the difference? The only difference is that the power of 2 to the power 3 whole thing we have multiplied with a factor which is 1 by 2 to the power 3; that means, N_y is this and this is called Q format.

So, here it is like $4Q.0$; that means, virtually there is no fraction; here it is like Q there is only one integer bit all fractions. So, it looks like $Q.13$. This is called Q format where there is no physical. You will only see b_3 to b_0 all like a number system because you are only accepting data which only takes between all like a 4 different bits and each bit can take only 0 and 1, that is it.

But, we want to represent how this output can be mapped to the analog voltage. So, for that N is nothing, but your quantized V a straight away and this is in the form of $Q.13$ it is an unsigned number. In the case of $Q.13$ you know the first bit is the MS first bit is MSB; if all bits are 1 because a first bit can take only 2^0 ; that means, it can be 1.

But, if all other bits are 1, that means, it is almost close to 2; that means, if you make all bits to be 1 it will be 2^0 plus 2^{-1} plus 2^{-2} plus 2^{-3} , how much it will be? $1 + 0.5 + 0.25 + 0.125$. So, it will you know at one point that the first last digit will be 5 next 2 to 4, 9. Sorry, the next will be. So, the next digit will be? So, last, the third digit is gone then $2 + 7 + 1 + 2^3 + 5$ 8.

So, that means, if all bits are 1, then you are getting a quantized voltage is 1.875 because; that means, your analog voltage let us say is 2 volts, but you are getting a quantized voltage of 1.875 volts; that means, there is an error of 125 milli volt which is pretty large may not be acceptable. So, you need to increase the number of bit sizes so that you can reach close to 2 volt. If we increase the resolution, then this number will further come close to 2 volt; that means, you take 6 bits then you will get another point another bit will come.

So, like that way you will approach, but again we discussed that too many bits are may not be acceptable, but what we understood from here that instead of writing all the time the actual binary we can write in terms of Q format and there is a standard way as if it gives you how to realize a binary number to a real-world voltage. So, you can say in $Q.13$ format in this case is a real number.

For example, if you are getting a bit; that means, let us say the first bit is 1 then 0 0 1, then in $Q.13$ what is the number? So, it will be simply 1; that means, 2^0 plus 2^{-3} . That means it is simply $1 + 0.125$. So, it is 1.125 volt; that means, this is a realization of this Q format, ok.

(Refer Slide Time: 13:40)

Concept of Q Format in Fixed Point Arithmetic

$Q_{n,m}$
 $n \rightarrow$ integer bit
 $m \rightarrow$ fraction bit

Stored
 $Q_{2,2} \rightarrow$ 4 bit binary
 integer bit 2's complement
 binary

-2 to 2

-2^{-1} 2^0 2^1 2^2 2^{-1} 2^{-2} 2^{-3}

$N_{num} = b_3 2^3 + b_2 2^2 + b_1 2^1 + b_0 2^0$

$N_{sig} = (-1) b_3 2^1 + b_2 2^0 + b_1 2^{-1} + b_0 2^{-2}$

Ex: 1 0 1 0
 $N_{num} = 2 + 2^{-1} = 2.5$
 $N_{sig} = -2 + 0.5 = -1.5$

So, this is the concept of the Q format. So, Q n dot 1 m; that means, n bits are integer m bits are fractions, but it is again a notional concept just to map a real-world number, but the actual total number of bit size is n plus m; that means, integer bit and the fraction bit. Again, this is a notion, but whenever we say Q format generally we by default consider sign because it is easy.

So, what will happen to the sign? Let us say we are talking about Q 2.2. So, Q 2.2 means we are talking about a 4-bit binary. So, this gives you a 4-bit binary. Now, it can be two things can be possible it can be straight binary or it can be a 2s complement. So, in straight binary in all cases we will write b 3, it is in the 2 2.2 formats, right b 3. I am talking about the straight binary, then what is that? Because we have 2 integer bits.

So, 2 integer bits mean the last bit must be 2 to the power 0. So, naturally, this will be 2 to the power 1, then b 2 2 to the power 0 plus b 3 2 to the power minus 1 plus b sorry, this will be; this will be b 1 and b 0 to the power minus 2, right? This is for straight binary, then what is what happens if that means, this is called straight binary I would say the unsigned number, but what is the signed number?

It will be minus 1 b 3 2 to the power 1 plus b 2 2 to the power 0 plus b 1 2 to the power minus 1 plus b 0 to the power minus 2. Now, we are talking about this. So, the example is that if we take let us say 1 0 1 0. In this case, what will happen? The unsigned number will be how

much the first digit is 1. So, there will be 2, then the third digit is 1. So, it will be 2 to the power minus 1. So, it will be 2.5, correct?

What will be the signed number because the signed bit is 1? So, it will be minus 2 plus 0.5. So, it will be minus 1.5, ok; so, that means, minus 1.5. So, that means, what are the 2.2 formats in the sign it can take a value between minus 2 to plus 2 in the case of an unsigned number it can take 0 to 4 close to 4. I am not talking exactly 0 to close to 2 because there will be a quantization error. So, this is what is the Q format.

And, from now onward we are moving with 2s complements. So, we are just talking about the second representation in this Q format.

(Refer Slide Time: 17:44)

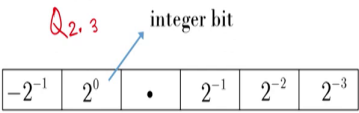

Concept of Q Format in Fixed Point Arithmetic

$Q_{n,m}$
 $n \rightarrow$ integer bit
 $m \rightarrow$ fraction bit

- Let $n = 2, m = 3$
- Let $N_x = 11010$ is a signed $Q_{2,3}$ number

$$N_{x(\text{dec})} = (-2)^1 \times 1 + (2)^0 \times 1 + (2)^{-1} \times 0 + (2)^{-2} \times 1 + (2)^{-3} \times 0$$

$$= -2 + 1 + 0 + 0.25 + 0$$

$$= -0.75$$



So, anything that we talk about you know this number it starts with let us say 2 point bit. So, this is like Q what is that? 2.3 format and by default let us assume we are talking about a signed number. So, then n equals 2 3. So, is a signed number, then as we discussed the first digit. So, it will be minus 75 sorry, it should be negative. No minus 1.2 minus sign is missing minus 75 because it is leading bit is 1. So, it must be negative ok.

(Refer Slide Time: 18:20)

- What are the largest and smallest values of N_X

$N_{X(\max)} = 01_111$

Q.2 MSB fractional part LSB
Q.n.m fictitious

Then what is the largest value of N_X in this case? What is the bit size? Again, it is in Q 2.3 format. So, this is the general representation of Q n dot m where n is the number of integer bits, and m is the number of fraction bits. But, in Verilog we can write underscore, this is for the notional concept that we are distinguishing between the decimal and integer. But, in Verilog this underscore will be ignored, it will treat simply this 5-bit number; but we need to use proper Q formatting to make it consistent as a fixed N number the first one is the MSB, then it is LSB.

(Refer Slide Time: 19:07)

- What are the largest and smallest values of N_X

$N_{X(\max)} = 01_111$

MSB LSB
Q.n.m fictitious

$N_{X(\min)} = 10_000$

MSB LSB

$N_X = b_4 b_3 b_2 b_1 b_0$

MSB LSB

So, this is like you know Q dot format minimum value. So, you can represent here MSB LSB as we have discussed it.

(Refer Slide Time: 19:15)

Addition in Q format

$$Q_{n_1, m_1} + Q_{n_2, m_2} = Q_{n, m}$$

$N_x + N_y$ $\max(n_1, n_2)$
 $Q_{1.4}$ $Q_{2.3}$ $Q_{2.4}$ $\max(m_1, m_2)$

$N_y = (-1)^{b_4} 2^0 + b_3 2^{-1} + b_2 2^{-2} + b_1 2^{-3} + b_0 2^{-4}$
 $N_x = (-1)^{b_4} 2^0 + b_3 2^{-1} + b_2 2^{-2} + b_1 2^{-3} + b_0 2^{-4}$
 $N_{x, \text{norm}} = (-1)^{b_4} 2^0 + b_3 2^{-1} + b_2 2^{-2} + b_1 2^{-3} + b_0 2^{-4}$

$N_{y, \text{norm}} = (-1)^{b_4} 2^{-2} + b_3 2^{-3} + b_2 2^{-4} + b_1 2^{-5} + b_0 2^{-6}$

$b_4 \ b_3 \ b_2 \ b_1 \ b_0$
 $= -b_4 + (b_3 2^{-1} + \dots)$
 $= -b_4 2 + b_4 + (b_3 2^{-1} + \dots)$
 $= -b_4 + (\dots)$

Now, we have discussed the Q format of a particular number, but suppose we want to add N_x plus N_y , both are signed Q formats, but they have different Q formats. So, in one case let us say this as Q 2.3, but if this has Q let us say 1.4 then how do we add it? So, we cannot add these two numbers unless we match their size, without matching we cannot; that means, first of all even though both are 5-bit numbers, we cannot simply add them because they are not in the same format.

So, we have to take for addition the integer bit; that means, it is like a Q. So, here in this case n_1 dot m_1 n_2 dot m_2 . So, we have to take a max of n_1 comma n_2 ; n_2 for m s. So, what is the maximum? There will be 2. So, you need to convert something like a 2 dot. Similarly, the max of m_1 comma m_2 and the max will be 4.

Now, the question is how can we convert this to this? Because if you remember this suppose this is my N_x . So, N_x will be something like how can I represent? N_x will be something like in 2 dot format, it is a significant number minus 1 b 2 2 to the power sorry, it is in the 5-bit number, sorry. So, it is a 5-bit number. So, it will be b_4 for this number.

Then 2 to the power since there are only two integer bits, it will start with this. Then b_3 2 to the power 0; then b_2 2 to the power minus 1; then b_1 2 to the power minus 2 b_0 2 to the

power minus 3 because there are three fractions b . Now, I want to convert; that means, I want to make $N \times$ normalization. So, this is in Q 2.3 format, I want to make it 2.4. How can you do that?

So, we can this number will not change if we write the same thing minus we maintain this 2 to the power 1 b 3 2 to the power 0 plus b 2 2 to the power minus 1 plus b 1 to the power minus 2 plus b 0 2 to the power minus 3 plus 0 into 2 to the power minus 4, it does not change. That means now this normalization will look like b 4 b 3 b 2 b 1 b 0 and 0. So, we have resized the data by padding one 0 or adding one 0; that means, we are adding a 0 in the LSB.

So, any data bit you add in the LSB with a 0 does not change the number. So, this number will remain the same. What will happen to this number? So, now, this format is Q 1 dot 4 formats I am just talking about $N y$. So, what does the $N y$ look like? Again, there is a minus sign it is a significant number you are talking about and this is again b 4 the first MSB because the number is b 4 b 3 b 2 let us say b dash just to avoid any confusion, then b 0 dash b dash.

Since there is only one integer bit it will be 2 to the power 0, then b 3 dash 2 to the power minus 1 plus b 2 dash 2 to the power minus 2 plus b 1 sorry, b 1 dash 2 to the power minus 3 plus b 0 dash 2 to the power minus 4. Now, I want to represent this number because I want to add this. So, this becomes a 6-digit number. Now, this is a 6-digit number where this is the fraction bit and this is the integer bit. Here we are putting a line, this is my fraction bit integer bit. Now, I want to represent.

So, I want to normalize y again normalize this is Q 1.4 format I want to convert it into Q 2.4, how do you do? It is simply a sign extension; that means, again it is a significant number minus 1. I will simply extend that bit by 2 to the power 1 and I the rest of the bit I will keep it; that means, b 4 dash 2 to the power 0, then b 3 dash 2 to the power minus 1 plus b 2 dashes 2 to the power minus 2 plus b 1 dash 2 to the power minus 3 plus b 0 dashes to the power minus 4.

So, by adding this 0, if you calculate mathematically you can show that this number remains the same because you know earlier this was the leading bit, ok. So, now, what we can do? So, here as you know; that means if you take any number; that means, this is called sign

extension; that means, a number which was earlier was $b_4 b_3 b_2 b_1 b_0$ is the same as another b_4 you can consider in case of the sign.

Because of what you are doing, you are adding another bit which means with this number we are because the second bit becomes now that bit has gone like because there now minus one has come out right here. So, we are extending the sign bit. So, now, this will become minus 2, then plus the other bit number; that means, what was this case? In this case, it was minus 1 b_4 that bit plus the rest of the number.

In this case, it will be minus b_4 into 2 because one extra bit is coming plus b_4 plus the rest of the number. The rest of the number means you know what is this. It will be b_3 into 2 to the power minus 1 all this dash sorry, dot dot dot dot b_3 dashes 2 to the power minus 1 dot dot dot. So, this part is common if you subtract you see this number after simplification will again become minus b_4 dash plus this number.

So, it does not change; that means, in 2s complement, we can simply extend the sign bit. So, we can resize this data by we can increase the MSB just simply extend the sign bit and that is a common practice. So, this sign bit extension does not change the value. Similarly, the extension with padding with 0 in the LSB, does not change the value.

(Refer Slide Time: 27:22)


Addition in Q format

$$Q_{n_1, m_1} + Q_{n_2, m_2} = Q_{n, m} \quad n \rightarrow \text{maximum of } n_1 \text{ and } n_2 \quad m \rightarrow \text{maximum of } m_1 \text{ and } m_2$$

- Rest of LSBs padded with zeros
- Rest of MSBs extended by MSB

Example: MSB
 Consider $Q_{2,2} = 10_11$ Convert it into $Q_{3,2} \Rightarrow Q_{3,2} = 110_11$

$$Q_{2,2}(\text{dec}) = -2 + 0.5 + 0.25 = -1.25$$

$$Q_{3,2}(\text{dec}) = -4 + 2 + 0.5 + 0.25 = -1.25$$


So, that means, we can resize this data in this format. So, that means, the maximum of this we have discussed rest of the LSB padding is 0, rest of the MSB is extended by MSB. Example

here. MSB converted into 3 dots 2. So, extend this b 2 with the sign bit, just now we discussed another sign bit being extended. So, it remains the same because mathematically you can show that another minus plus will get canceled. So, it will minus it remains same.

(Refer Slide Time: 27:49)

Multiplication of Signed Numbers in Q Format


$$Q_{n_1, m_1}^{\checkmark} \times Q_{n_2, m_2}^{\checkmark} = Q_{n, m}$$

$$n = n_1 + n_2, \quad m = m_1 + m_2$$

However $Q_{n, m} \rightarrow$ consists of 2 signed bits, which can be modified as

$$Q_{(n_1+n_2-1)(m_1+m_2+1)}^{Q_{n, m}}$$

one extra bit can be used for further
progressing with improve resolution



Now, if you want to multiply you can multiply with two different formulas, does not matter. So, this will become what will be multiplication n will become n 1 plus n 2; m will become m 1 plus m 2. But, since this already has a sign bit. So, the resultant will have two sign bits; that means, there is one extra sign bit that is redundant and I can simply discard the sign bit and I can do a left shifting operation and pad with a 0.

That means what I am meant to say consists of 2 sign bits. So, I can eliminate one sign bit; which means, I can do the shifting, and then for another LSB I will just pad with 0 so that this will; means, it should be plus 1 to keep the same Q n dot m format. The data size will remain the same the value will remain the same, but you are giving an extra headroom for the LSB so that when you do the next operation this extra headroom can be used for final resolution. So, it is giving 0, this is the sign bit. So, we can discuss it.

(Refer Slide Time: 29:00)

Co-efficient Scaling in Fixed Point – Example of Digital PI Controller

Write discrete-time PI controller

$$N_{con} = N_{prop} + N_{int}$$

$$N_{int} = N_{int,prev} + K_i \times N_x$$

$$N_{prop} = K_p \times N_x$$

$$u_x(n) = u_x(n-1) + K_e(N_x)$$

Identify different Q formats

$K_i \rightarrow Q_{1.9}$

$K_p \rightarrow 0 \dots 11111111$

$K_{p(max)} \approx 8$

2^{-5}

$= \frac{1}{32}$

2.03

2.06

So, extra bits can be used for further processing for higher resolution. Now, we want to why do we need all this. Suppose, we are talking about the PI controller ADC. Let us say this is the error voltage and this is the error signal and we need to process that. So, what if I take the first proportional control? What is my proportional control?

Suppose, I am talking about a proportional control where this is I am talking about a K_p into N_x . Let us say N_x has $Q_{1.9}$ in sign. What does it mean? That means, this voltage varies between minus 1 volt and 2 plus 1 volt, and $Q_{1.9}$ means the maximum value can be plus 1 close to plus 1 minimum will be minus 1. So, it is perfect. Now, for the proportional game, I want to vary between 0 to let us say you know 3. So, I am taking Q to be 4 points let us say 5.

4 bit means I leave one bit 0 because the proportional gain cannot be negative. So, one bit is gone you have 3 bits. So, you can get up to 8 using 3 bit; that means, almost close to 8; that means if you keep all bit; that means, it is a 9-bit K_p . So, that means, if you realize 0 and all one because there are total 9 bit 1 2 3 4 5 6 7 8; the first bit is 0, then all 1. So, this will give the maximum K_p . So, the K_p max will be approximately equal to 8.

What is the K_p mean because we are not talking about negative K_p , it can be 0. All bits are 0. So, we can vary the proportional gain between 0 to 8, but at what resolution? So, this will be defined by this type. So, the resolution will be 2 to the power minus 5 ; that means, 1 by 2 to the 32 . So, it can vary in the scale of almost 0.3 0.03 , 30 milli; that means, you can vary between 2.03 then 2.06 like that, ok. So, in that way, you can set a proportional bit.

So, now in the PID controller, the overall output will be N proportional plus N integral. So, the proportional part is clear; that means, we got proportional what is the format of this guy after resizing so? That means, if we do not resize for the time being it will be Q because we know this N 1 N 2 M 1 M 2. So, it will be Q 5.14 because 5 M 1 14. So, we got Q 5.14.

Now, we are talking about integral gain. What is integral? We know that integral has integral previous plus there will be K_i into $N \times$ instantaneous term. If we know in the actual equation we know u_i of n is equal to u_I of n minus 1 plus K_i into V error n . So, something similar you are writing. But, now I want to understand what is the size of this guy.

So, that means, K_i what we know $N \times Q$ format is Q 1.9 because it is defined by this ADC then what will be the Q format of K_i ? First of all K_i cannot be negative. Now, in this, it is talking about discrete time K which is typically less than 1. So, it is reasonable that we can take 1.9 wherein in a positive sense it can be a maximum of 1 close to 1.

If all of the first bit is 0 the MSB is 0 and all are 1 it will be close to 1 because we cannot take negative K . What can be minimized because this K_i will be a fractional number. So, you have to give sufficient resolution on the right side, but after this multiplication K_i into $N \times$. So, we are getting the resolution of 2.18, first of all, we need to resize it because there is an extra MSB here sign bit extra sign bit. We need to resize this there is an extra sign bit here. So, this size is what we needed.

We cannot simply multiply this and add these two because they are in a different format and we have discussed how to resize it first of all we need you to need to identify the Q format. So, this will give a headroom what is the variation of integral gain that we want. So, we have identified ok.

(Refer Slide Time: 34:23)

Co-efficient Scaling in Fixed Point – Example of Digital PI Controller

Summary of steps for arithmetic operation

$$N_{con} = N_{prop, nom} + N_{int, nom}$$

$Q_{n.m} \quad Q_{n.m} \quad Q_{n.m}$
 $Q_{n.m} \quad Q_{n.m} \quad Q_{n.m}$


V_{min} to V_{max} → ADC → N_x

$N_{prop} \rightarrow Q_{5.14}$

$N_{int} \rightarrow Q_{2.18}$

$N_{int, nom} \rightarrow Q_{5.18}$

$N_{prop, nom} \rightarrow Q_{5.18}$



Next the summary; that means, the summary is that N is the control that is one, what is the format? It has to be m proportional nominal because we need to add in a proper format N integral nominal. What do the nominal? They have a compatible Q format otherwise you cannot.

So, that means, we are talking about n dot m; this must be n dot m. What is n, and what is m? That we will discuss and if we add to the number it can be the same because in the worst case what will happen if both numbers are maximum positive there can be overflow. So, we have to make sure because if you add two binary numbers, a 4-bit number, and a signed number, if both are positive then there is a possibility of overflow and that will be detected because, in a significant number, you have to be very careful about whether there can be rapping error.

If both are negative they also have to check otherwise if both have an alternative sign or they are not maximum value, then you may not have any kind of overflow. So, typically they are in the same number, but we need resizing of the data because we got actual proportional gain in the format of Q if you go back to our previous 5.14 and integral in the form we got 2.18.

And, remember if you go back to this integral previous if it is Q 2.18 this will be 2.18, this will be 2.18 in summary. Again, we are adding the two same-size data we should make sure that this data and there is a high possibility integral that it can saturate. So, when you go to the integral control we will take care of the saturation limit.

So, we need a resizing. Again, what will be the minimum size so the 5 has to be maintained? We cannot discard any MSB. So, we need integral normalization you have to make it 5.18 so that we extend with sign bit 3 extra sign bit, and similarly, for proportional control we need to extend the Q 5.18 where the remaining 4 bits will pad with 0. So, that means the sizing that is 1. So, this is the addition of 2.

(Refer Slide Time: 37:13)

Co-efficient Scaling in Fixed Point – Example of Digital PI Controller

Resizing data in digital PI controller

V_{\min} to V_{\max} → ADC → N_x

$Q_5 N_{10^8}$

And, next, you have to resize the FPGA digital data to get the controller output should be in the form of let us say so; that means, we got let us say what we got 5.18.

(Refer Slide Time: 37:36)

Co-efficient Scaling in Fixed Point – Example of Digital PI Controller

Resizing data in digital PI controller

V_{\min} to V_{\max} → ADC → N_x

$Q_{5.18}$
 N_{10^8}
Saturation

3 bit signed counter

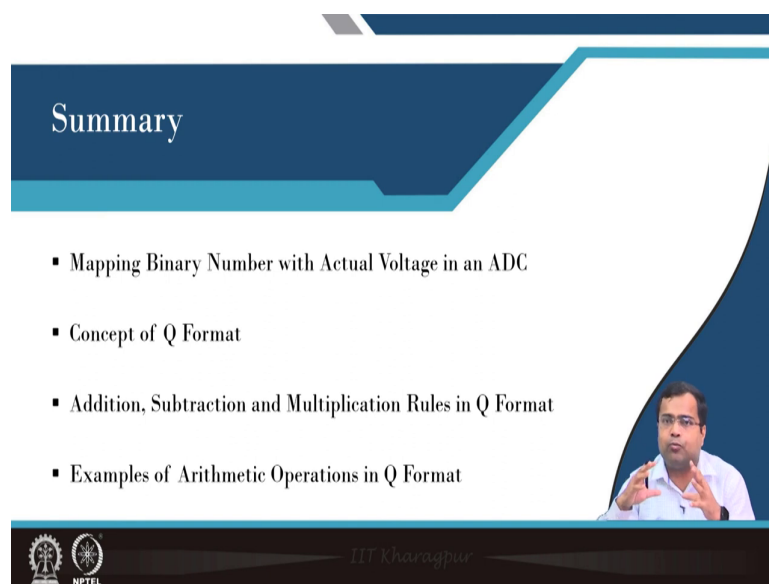
011
100

So, we need to get 5.18 in this format; that means, we got 5.18 as the controller nominal. But, this may be too large because what will do with this control output? So, this control output has to be compared with a short waveform, but we have to check what is the resolution of how we will go to that point; that means, we need also further resizing.

So, first of all, we need resizing of the integral term, and proportional term to add them, and then we need resizing of the Q to fit because you have to compare with another signal that is one and. Secondly, we also need a saturation block because there is a high possibility that the integral will saturate and there is a sign so, there can be a wrapping error. That means if you take a counter let us say you are taking a 3-bit signed counter, what will happen?

So, if 0 wall one reaches the next you hit it will be 1 or 0. So, this will be a rapping error and that can cause a huge problem in your controller. So, you have to be very careful we can put a limit here so that it should not exceed. So, similarly for the controller so, we need to put a limit.

(Refer Slide Time: 38:55)



Summary

- Mapping Binary Number with Actual Voltage in an ADC
- Concept of Q Format
- Addition, Subtraction and Multiplication Rules in Q Format
- Examples of Arithmetic Operations in Q Format

IIT Kharagpur
NPTEL

So, in summary, we have discussed binary numbers with actual voltage, discussed the Q format, we discussed how to resize the Q format for addition, subtraction, and multiplication, what is the law, and then how to do the arithmetic operation in Q format some basic idea we have discussed; that means, processing of the resizing.

That is it for today. Thank you very much.

