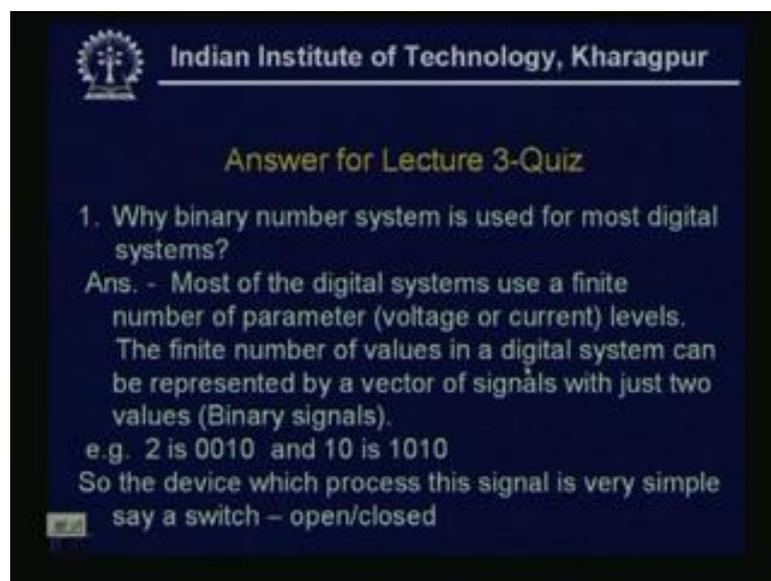


Digital System Design
Prof. D Roychoudhry
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 04
Digital Logic II

May, I before starting the today's lecture first we quickly go through the last day's lectures answers of the quiz.

(Refer Slide Time: 01:14)



Indian Institute of Technology, Kharagpur

Answer for Lecture 3-Quiz

1. Why binary number system is used for most digital systems?

Ans. - Most of the digital systems use a finite number of parameter (voltage or current) levels. The finite number of values in a digital system can be represented by a vector of signals with just two values (Binary signals).
e.g. 2 is 0010 and 10 is 1010
So the device which process this signal is very simple say a switch – open/closed

The first question was why binary number system is used for most digital system, now as you know the most of the digital systems use discrete number of parameter levels say the voltage levels or current levels. And, the finite number of values in a digital system can be represented by a vector of signals with just two values, these are binary signals means that 2 that can be represented by a string of 0 and 1 0 0 1 0 or 10 that is 1 0 1 0. And the device which process the signal is very simple to implement say a switch either 0 or 1 means switch on or so off or open or closed. So, that is why that mainly binary number system is used for more most of the digital systems.

(Refer Slide Time: 02:03)

Indian Institute of Technology, Kharagpur

Answer for Lecture 3-Quiz

2. Convert the following binary numbers to decimal
(a) 10101.001 (b) 110010.0101

Ans. –

(a) $10101.001_2 = 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$
 $+ 0 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3}$
 $= 21.125_{10}$

(b) $110010.0101_2 = 1 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0$
 $+ 0 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} + 1 \cdot 2^{-4}$
 $= 50.3125_{10}$

Second question was the conversion, the following binary numbers to decimal. So, as here the binary is in binary radix is 2 using the formula we did this should be 1 0 1 0 1 means 1 into 2 to the power 4, 4 is the place value. Similarly, 0 into 2 to the power 3, plus 1 into 2 square 0 into 2 to the power 1 1 into 2 to the power 0.

And the non-integer part, 0 into 2 to the power minus 1, 0 into 2 to the power minus 2, 1 into 2 to the power minus 3, so if you do this computation we have getting 21.125 in decimal number system. Similarly, the other one also similarly you can get that it is 50.3125 in decimal number systems.

(Refer Slide Time: 03:17)

Indian Institute of Technology, Kharagpur

Answer for Lecture 3-Quiz

Convert the following decimal numbers to binary
(a) 678 (b) 129.84

Ans. –

| (a) 678_{10} | remainder | | remainder |
|----------------|-----------|-------------|-----------|
| $678/2 = 339$ | 0 | $21/2 = 10$ | 1 |
| $339/2 = 169$ | 1 | $10/2 = 5$ | 0 |
| $169/2 = 84$ | 1 | $5/2 = 2$ | 1 |
| $84/2 = 42$ | 0 | $2/2 = 1$ | 0 |
| $42/2 = 21$ | 0 | $1/2 = 0$ | 1 |

$678_{10} = 1010100110_2$

Third question was the convert the decimal numbers to binary. So, the first one was 678 in decimal in this case we will divide by 2 as I want the number to be converted in binary system where the radix is 2. So, divide by 2 and this will be a repeated division and we are accumulating the remainders, 678 divided by 2, remainder is 0.

For the second division remainder is 1, similarly I am getting this remainders, again it is continuing here 21 divided by 2, actually it will be remain remainder is 1. Then, this next remainder is 0, next is 1 like that, and what we will be getting the answer, this is 678 in decimal number system, 10 is that we have to take the remainder in reverse way; that means, 1 0 1 0 1 0 0 1 1 0, so this is my in binary.

(Refer Slide Time: 05:08)

Indian Institute of Technology, Kharagpur

Answer for Lecture 3-Quiz

Convert the following decimal numbers to binary
 (a) 678 (b) 129.84

Ans. –

| | |
|---------------------------------|---------|
| (b) $129_{10} = 10000001_2$ | integer |
| $0.84 \times 2 = 1.68$ | 1 |
| $0.68 \times 2 = 1.36$ | 1 |
| $0.36 \times 2 = 0.72$ | 0 |
| $0.72 \times 2 = 0.44$ | 0 |
| $0.84_{10} = 0.1100_2$ | |
| $129.84_{10} = 10000001.1100_2$ | |

And, the b part answer of the b part is there is one non-integer portion also, so the integer portion is similar by repeated division we will get that then integer portion that 129 in decimal number system is 1 0 0 0 0 0 1. We can see this thing in other way see that actually the place value here this is 0 1 2 4 8 16 32. So, this is the places see if we take the here this value will be 128 and this value will be 1, this is 128 plus 1, because rest of the digits are 0. So, this is 129 in decimal number system.

Now the non-integer portion, I can get by repeatedly multiplication by the radix value. So, this is the first non-integer is 0.84 it will be multiplied by 2, then it will be 1.68 and integer is 1. Similarly, I will get again the non-integer portion multiplied by 2, another integer 1, I am in that way I am getting that this is 0 this is 0. So, the non-integer portion

0.84 in decimal it will be 0.1100 in binary. So, the number 129.84 in decimal is converted to binary and the binary number is 1000000 1.1100 and this is approximated.

(Refer Slide Time: 07:57)

Indian Institute of Technology, Kharagpur

Answer for Lecture 3-Quiz

4. Convert the following decimal and binary numbers to BCD

(a) 5648 (b) 10001101

Ans. -

(a) $5648_{10} = 0101\ 0110\ 0100\ 1000$ (BCD) ✓

(b) $10001101_2 = 141_{10} = .0001\ 0100\ 0001$ (BCD)

Now, the last question was the conversion from decimal and binary numbers to binary coded decimal the BCD value. Now, see that the first number was a 5648 in decimal, so the way we have define the binary coded decimal we will take that for each decimal digit that four such binary values. So, that means, here 5 means say this is a 0101, 6 means this is a 0110, 4 means 0100, 8 means 1 0 0 0, this is my BCD.

Now, the part b is the binary number, already it is a binary number is given this is 1 0 0 0 1 1 0 1 in binary. Now, what is the decimal equivalent of this, because this is 1 this is 4 8 16, 0 into 16, 0 into 32, 0 into 64, this is 1 into 128; that means, the value is 141 in decimal. Now, once it is converted into decimal, now the same way that each digit is replaced by 4 binary digits; that means 1 means 0 0 0 1, 4 is 0 1 0 0 and 1 is again 0 0 0 1, this is my binary coded decimal value. So, these are the answers of the lecture 3 quiz.

(Refer Slide Time: 10:05)

Indian Institute of Technology, Kharagpur

Accuracy of Binary Conversion of Non-integers

- When a non-integer decimal is converted to binary the maximum error in decimal is 2^{-n}

A binary number is expressed as $0.1010ijk\dots$
i, j, k can be 0 or 1

If 5th bit and all succeeding bits are 0 then the number is accurate up to 4th bit position
But if 5th bit and all succeeding bits are 1 then the error would be $E_2 = 0.00001111\dots_2$

Now, we start our today's lecture the digital logic 2, first we see the accuracy of binary conversion of non-integers. See, when a non-integer decimal is converted to a binary number, sometimes we truncate the number of digits by n say n can be 4; that means, after decimal points, I can take 4 digits I can take 5 digits or say 9 digits. But still, it does not stop; that means, that fraction part does not becomes 0, but we stop, this is called that truncated to some n digits.

Then; obviously, there is there will be some errors, because if we increase the number of bits, then we will be nearer to the actual value we will be near and nearer to the actual value. So, now this is called we call this we are defining as a this is a error; that means, if I take up to after decimal points, if I take after 5 digits, I will be getting a more accurate result, instead if I take three digit value.

Now, the maximum error in decimal is defined as 2^{-n} to the power minus n, if it is truncated by n bits. See, why I am we are telling this thing, see a binary number is expressed as say 0.1 0 1 0 i j k l like that; that means, say here after decimal point I am taking 1 2 3 4 5 6 7 say 7 such bits, i j k can be 0 or 1.

Now see, as this is a it is converted to a binary number systems, that is why i j k are 0 or 1. Now, up to 4 bits already the 1 0 1 or 0 value is there. But, from the 5th bit onwards I have represented the bits as i j k like that; that means i j k came be either 0 or 1. Now, if the 5th bit and all succeeding bits are 0, then the number is accurate up to 4 bit position, because these are remaining bits are all zeros. So, what I can tell that this is 0.1010, then

we have all are zeroes no need to write this thing. So, I can tell that this is 0 point even it is up to 4th or here in this case it is up to 4th bit position this is correct.

Now, see if 5th bit and all succeeding bits are 1 then the error would be that point 0 0 0 0 and then these are all ones. So, actually these are not zeroes, if it is 0, we can easily discard those values, but if i j k these values are 1, then I cannot discard that, then actually I am approximating that values. So, there must be some errors introduced in the binary number.

And, what is that error, and that error is that up to 4 decimal point it was the value was given in binary. Now, I am taking as it these are all 0 and this i j k values and succeeding values are 1, so as actually these are the, it were these values, where actually present or it should be incorporated in the binary number. But, we have dropped this value from the binary number.

(Refer Slide Time: 14:15)

Indian Institute of Technology, Kharagpur

Accuracy contd.

- In decimal, the error is

$$E_{10} = 2^{-5} + 2^{-6} + 2^{-7} + 2^{-8} + \dots$$

$$= 2^{-5} (1 + 2^{-1} + 2^{-2} + 2^{-3} + \dots)$$
- The term in parentheses is a geometric series, it approaches a value $= 1/(1 - 1/2) = 2$
- $E_{10} = 2^{-5} \cdot (2) = 2^{-4}$
- So, if n bits are used to represent the non-integer part of a binary number, the maximum error due to termination is

$$E_{10} = 2^{-n}$$

So, that is why it is a, this is defined as a error, now we see that, how we have approximated this thing. See in decimal the error is E 10, how it is coming, is 2 to the power minus 5 2 to the power minus 6 2 to the power minus 7 plus 2 to the power minus 8. We see the previous slide again, see actual I have dropped the 5th bits and the all succeeding bits and if they were all once then it was we have define these as the error.

Now, see this is the place value, this is after decimal point this is the 5th bit, so it will be 1 into 2 to the power minus 5. The second 1 is 1 into 2 to the power minus 6 and that I was getting here. See that this is 2 to the power minus value the 5th bit this is the error

for the 6th bit 1×2^{-6} to the power minus 6 2^{-7} to the power minus 7 2^{-8} and so on.

Now, if I take common that 2^{-5} , then it will be $1 + 2^{-1} + 2^{-2} + 2^{-3}$ like that. Now, this term within the parenthesis this term is a geometric series and it approaches a value nearly equal to $1 / (1 - 1/2)$ and which is nothing but the 2. So, this value this within the parenthesis this value becomes this is becomes 2.

And if you, what will be the value of E_{10} , this is 2^{-5} and multiplied by 2, this value is 2^{-4} . So, if n bits in our example, we have tried n equal to 4; that means, up to 4 bits the binary number was given or the number was truncated up to 4 bits after decimal point the 4 bits I have taken.

So, if n bits are used to represent the non-integer part of a binary number, then the maximum error due to termination is E_{10} equal to 2^{-n} ; that means here, 4 is replaced by n in general cases. So, this we are telling that accuracy of a number when it is converted from decimal to binary.

(Refer Slide Time: 18:01)

Indian Institute of Technology, Kharagpur

Binary Coded Octal (BCO)

- For Octal number system the base or radix is 8
- The eight symbols are 0 through 7
- The computer uses Binary Coded Octal
- Three binary positions are used to represent one digit of octal number

| | | | |
|-------|-----|-----|-----|
| Octal | 2 | 3 | 7 |
| BCO | 010 | 011 | 111 |

Handwritten notes:
 2-010
 2-011
 10 011 111

• Advantage – ease of conversion to and from binary

Now, last time we have read that, what do we mean by octal numbers, what do we mean by hexadecimal numbers or octal number systems hexadecimal number systems etcetera. Now, today we see another type of binary number systems, what we call that binary coded numbers, why, because already we have mentioned that the mainly the computer

handles or it very advantageous if the it is all the numbers that are used in computer is represented in a binary number systems.

So, even it is a used in octal or hexadecimal, how they are converted into strings of zeroes and ones, means mainly the binary values only two values. Then obviously, it will be very much it is very easy to implement in for digital systems, just the simple example. I have given that one switch or else we can model as a switch on is one switch off is 0; that means, this is the binary system.

So, first we see that what is binary coded octal, normally called BCO. So, for octal number system already we have seen that the base or radix is 8 and the 8 symbols are that 0 to 7 means 0 1 2 3 4 5 6 7 these 8 values are mainly used. Now, the computer uses binary coded octal, already I have mentioned, now how this octal number is converted to binary coded octal, that say three binary positions are used to represent one digit of octal number.

Let us take an example, say the example I have taken, this is octal is 2 3 2 3 7 that is in octal 1. Now, each positions say this positions that second position value is 2 in octal, that will be represented by 3 bit binary means say two means. If I take that 2 binary of 2, how I can do that thing, that again if I it will it should be divided by 1, then it will be 1 0.

So, if I take this value this will be 1 0, so if I want to represent 2 by 3 bit binary numbers this will be 0 1 0, the leading bit I have given 0, it is of no meaning so 010, so the BCO of 2 is 0 1 0. Similarly, the 3 we can represent as 0 1 1 and the 7 is 1 1 1, the octal 2 3 7 8, that gives the binary coded octal that 0 1 0, 0 1 1, 1 1 1.

Now, it is very much advantageous, if I want to convert that thing to something binary and what we will be doing, that we will suppress the leading zeroes. See, here only one leading zeroes are there, see only this is 0, so the binary will be value will be only 1 0 0 1 1 1 11. So, that advantage is that it is very easy to convert from binary to BCO or BCO to binary and that is why these are very much used in the computer systems.

(Refer Slide Time: 23:12)

The slide features the IIT Kharagpur logo and title at the top. Below, it lists three conversion examples with handwritten annotations:

- $159_{10} = 1011111_2 = 001\ 011\ 111 = 137_8$ (Handwritten: 137₈ above the result)
- $67258_{10} = 110\ 111\ 010\ 101$ (BCO) (Handwritten: 67258₈ above the binary groups)
- $2.51_8 = 010.101001$ (BCO) (Handwritten: 10.101001₂ below the binary result)

Now, again some other different examples we take that this is 159 in decimal number systems this is 159. So, first this is converted into the binary that 1 0 1 1 1 1. Now, what I can do that as it is a binary, actually I am doing the reverse thing now binary to BCO. So, what I we will be doing, we know that the three digits I have to take together, so this is from if I start from the LSB side from the right hand side to left, then this 3 is grouped this 3 is grouped and see only 1 1 is left. So, what I can do, I can put say 2 zeroes here, because in certain 2 0 in the leading position does not effect, because it means that value is 0.

So, how I group in 3 bits, this is 0 0 1 this is 0 1 1 this is 1 1 1, now if I convert 0 0 1 means we know that this is 1 this is not true. This is actually 0 0 1 means this is 1, then 0 1 1 means this is 3 and 1 1 1 means this is 7. So, this is my 137 in octal value. Now, if I take another example 67258 in decimal number system, then first it is converted to say that decimal binary numbers. So, this is 1 1 0 1 1 1 0 1 0 1 0 1, we are taking this as the octal value. So, this is 6725 and the base is 8.

Now, what we will be doing, again that each digit in octal that should be represented by three binary bits. So, the 6 this is nothing but 6 represented in 3 binary bits this is 7, this is actually 2 and this is 5, this is my BCO binary coded octal, what will be binary, because there is no leading 0, no leading zeroes are there.

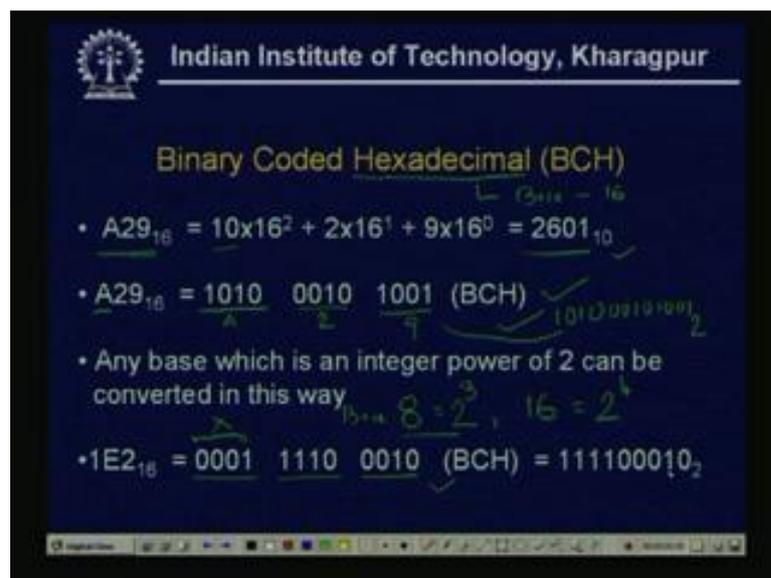
So, if I just merge this 3 bit numbers, then it will be my binary that 1 10 1 1 1 0 1 0 1 0 1 this is my binary numbers. Now, we consider one fraction, this is 2.5 in octal, then what

we will be doing as, this is the there are two portion, this is my integer portion this is my integer and this is my non-integer portion. So, in integer portion we will be doing the 2 is represented by 3 binary bits 0 1 0.

Now, the non-integer portion will be an again similarly can represented a 5 means 1 0 1 and 1 means 0 0 1, so this is my binary coded octal and what will be the binary again the one leading 0 is there. So, we just drop the leading 0 and we merge, so what this is 1 0.1 0 1 0 0 1 in binary.

So we see that, how that one octal value can be converted to one binary coded octal value and from there that binary coded octal value to binary. So, what we can tell that octal to binary or BCO binary coded octal to binary and the advantage is that the ease of conversion and that is, why that BCO is very popular.

.(Refer Slide Time: 28:49)



Now, the similar number system is the binary coded hexadecimal, because similar to octal, already we have read another number system that is hexadecimal number system. And there that, hexadecimal means the base is here the hexadecimal the base is 16 and we have seen actually 0 to 15 values from 10 to 15 or actually represented by 10 means A, 11 is B and similarly that 15 means F. So, 0 to 9 and A to F this 16 digits are the symbols of hexadecimal system, already we have read in the last class.

Now, we see that how this binary coded hexadecimal; that means, how this hexadecimal numbers are converted to into binary coded numbers. This is very similar to our BCO the

binary coded octal, again we take one example, so this is my hexadecimal numbers A 2 9, so A will state.

So, if I want to convert that thing in decimal that 10 into 16 into square because A position is 2, 2 into 16 to the power 1 plus 9 into 16 to the power 0, which becomes 2601 in decimal number system. Now, the same number if I want to convert in BCH, now in this case each hexadecimal digit will be represented by 4 bits, these are everyone will be represented by 4 bits in octal it was represented by 3 bits.

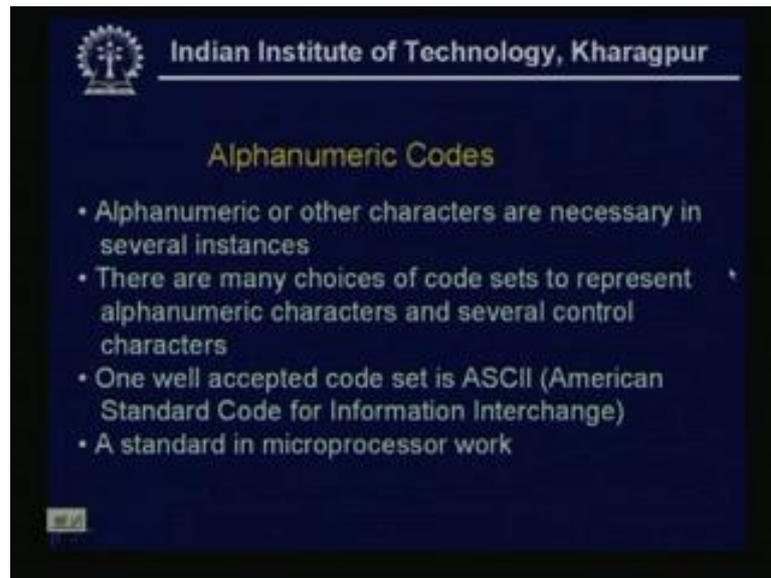
So, this A means 10, this is 1 0 1 0 that 10, 2 this is my 2 and this is my 9, this is the BCH code or binary coded hexadecimal code. And similarly, if I want to convert this thing by binary, then I will just drop the leading zeroes and merge all the values. Here, there is no leading zero, so it will be 1 0 1 0 0 0 1 0 1 0 0 1, this will be the value.

So, any base which is an integer power of 2 say 8 means 2 to the power 3; that means, octal when base is when base is 8 that is a power of integer 2 or 16 when base is 16 for hexadecimal occurs, this is power of an integer 2 4th power of 2. Then, it can be converted in this way, this is the very simple way, but the only restriction is that if the radix or the base is a power of integer 2.

So, if we see another example that 1 E 2 in hexadecimal number, then that this is 1 is represented by 4 binary bits E means that 14 is represented by 4 bits and 2 is represented by 4 bits this is called BCH. And what will be the binary, see here there are three leading zeroes, so drop this 3 zeroes, then it will start from 1 1 1 1 0 0 0 1 0, so this is my binary number.

So; that means, from 16 hexadecimal number we have converted to BCH and from BCH to binary again this is very simple just drop the number main leading zeroes and merge the other binary values. So, this will give you the binary number.

(Refer Slide Time: 33:44)

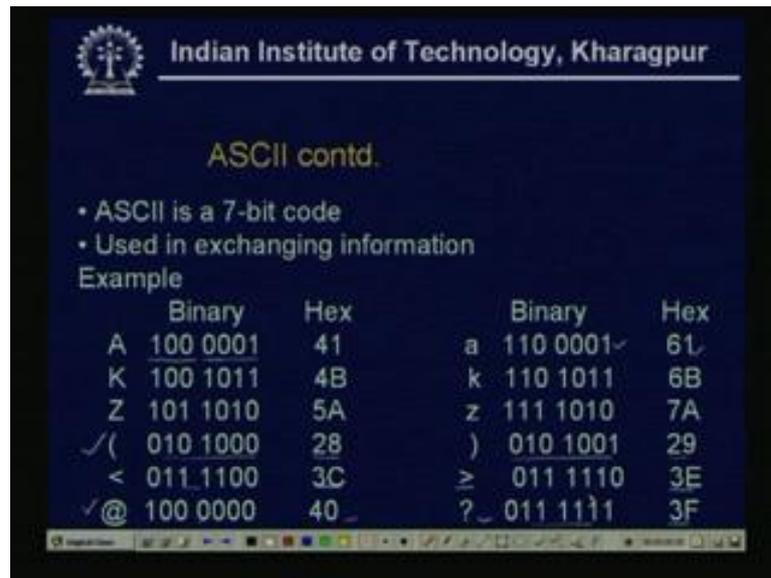


Now, we see that some other codes are called the alphanumeric codes, now alphanumeric codes or other characters are necessary in several instances. So far, we have seen only the numbers, either it can be decimal number either it can be octal number or it can be hexadecimal numbers. But sometimes we need characters and some other control characters.

Then, how they can be converted into binary values 0 or 1 if it is used in computer see in our keyboard all of we know that keyboard that A to Z, the 26 characters are there 0 to 9, the 10 decimal values are there. And, there are say some more control characters say some mathematical operators are there less than greater than some question mark is there. Then how this characters or this alphanumeric characters, how they can be converted into 0 and 1.

Now, there are many choices of code sets to represent alphanumeric characters and several control characters. Now, one will excepted code set is the ASCII and the full term form is the American Standard Code For Information Interchange, so this is a very well known standard in micro processor work.

(Refer Slide Time: 35:51)



So, we can see that how this, what is this ASCII code, see ASCII is a 7 bit code normally used in exchanging information. See in our keyboard that when we press a A., so actually A means immediately it is converted to a 7 bit binary numbers. This is predefined this is a predefined numbers say this 1 0 0 0 0 1, this is my capital A.

If I represent this thing in hex will be simple, because just now we have done binary to hexadecimal means we have to group it will be grouped by 4, here it is 4 and here another 0 I will put. So, 0 1 0 0 means this is 4 and 0 0 0 1 means this is 1, this is a hex 4 1. Similarly, the small letter that will be by this 7 bit value binary values that 1 1 0 and 1 1 0 1 and this is my hex is 61.

Similarly, that all A to Z are given and a 2 3 examples that capital K capital J, how they are defined, so; that means, here the else these are 7 bit ASCII. So, 2 to the power 7 mean that 128 such characters and control characters together. Total 128 numbers I can represent by this 7 bit binary values and always that just this is easy to express the hexadecimal because it is very smaller in size. So, the hexadecimal values are also given.

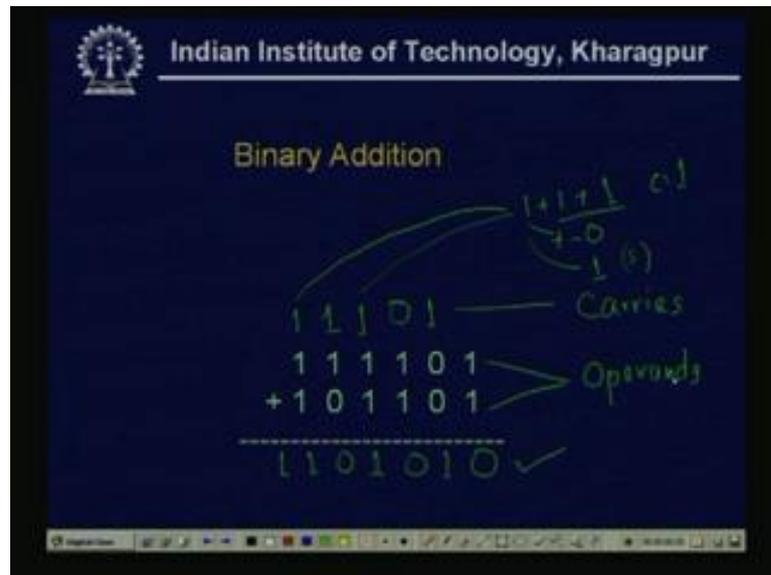
See this is the parenthesis, that opening parenthesis is given by these 7 bit binary values and the hex is 28. The closing parenthesis similarly it is define this 7 bit numbers and this hex value is 29, if it is an operator say less than or it is a greater than then also that it has some ASCII code.

So, we are telling this is the ASCII code of this 7 bit binary or the hex, these are the ASCII code of less than or greater than say at the rate. So, this is the 7 bit code and hex

code is form 4 0, similarly the question mark the ASCII code of question mark is 3 F of this is a numbers.

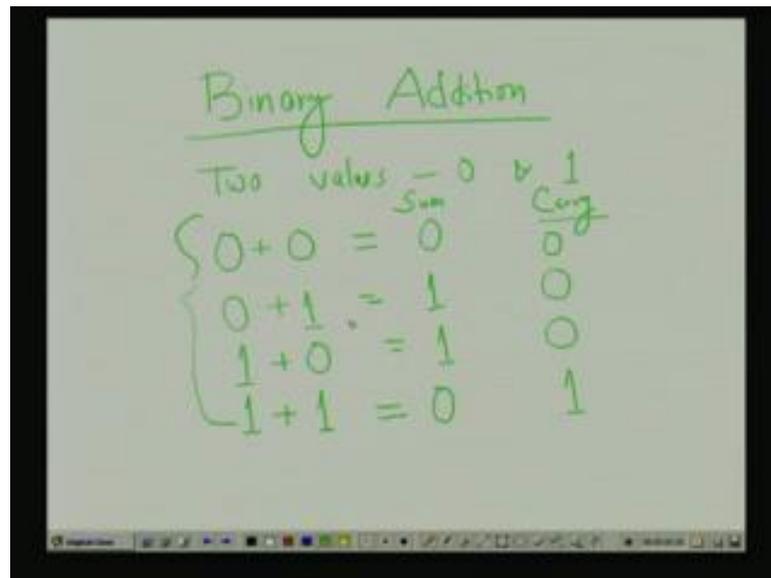
And. So for, that in the keyboard whatever characters and the control characters the mathematical operators say some other special functions say F 1 F 2 F 6 some del acknowledge or home end everyone that has some ASCII code. So, when we are pressing this thing immediately it is converted into that 0 1 values or strings of 0 ones these are predefined and obviously, it will be then realized as that particular A.

(Refer Slide Time: 39:28)



Now we see, another three basic operation in binary system, because so far from our discussion, what we have seen that mainly in computer or in digital systems, we will be using the binary number system. Now, in binary we have only two values because binary the base is 2, so it is the 0 and 1 value, so there are only two values that 0 and 1. So, first thing I have to define if I want to define a binary addition.

(Refer Slide Time: 40:27)



So, first how this binary digits are added, so as there are two values there can be seen in binary addition say we have two values 0 and 1 0 and 1. So, what are the different addition of digits 1 is 0 plus 0 and we are defined this as a 0 sum is 0, I am telling sum is 0 and on from our normal sense that carry is also 0; that means, if we add 0 and 0 the sum is 0 carry is also 0.

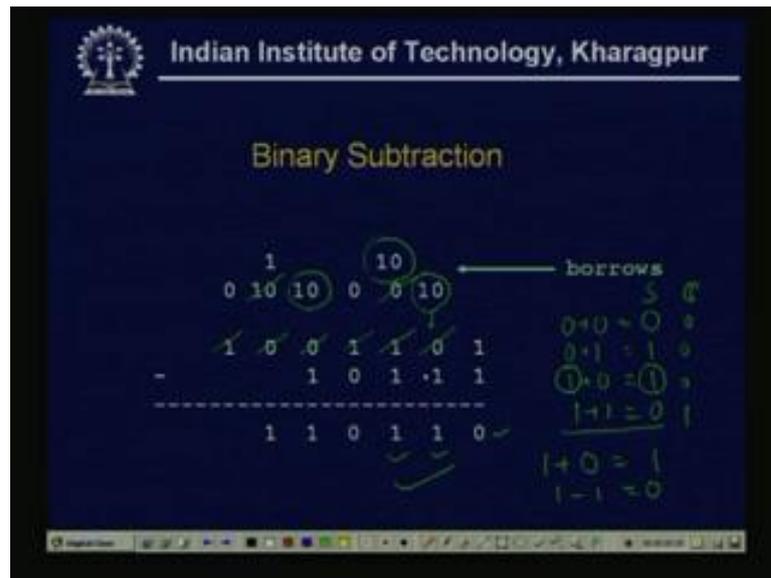
Now, if I add 0 and 1, then sum is 1 carry is again 0 and this is same, because it is addition, so 0 plus 1 and 1 plus 0 is same and the sum is again the 1 carry is 0. Now, if it is 1 and 1, because there are only two digits in the binary system, so these are the four such different additions is possible. So, this is 1 plus 1 then sum is 0 carry is 1. So, if I know these basic additions of binary digits, then I can do any binary addition whatever large it is, but it is very simple.

We see that thing, see I have taken 1 example, that these are my two operands that 1 1 1 1 0 1 and I am adding that thing with 1 0 1 1 1 1 0 1. So, if it is 1 1. I am adding that means, my sum should be sum should be 0 carry should be 1, then 0 plus 0 that is 0 sum is 0 and again that is added with another 1. So, that sum is 1, but carry is 0.

Now, again here see that this is 1 plus 1 means my sum is 0 and that but carry is 1 and again that sum 0 plus 0, this is 0 carry is 1. Now, we see these values that, so actually this is 1 plus 1 that sum is here for this I am getting 1 plus 1 plus 1. So, here 1 plus 1 means sum is 0 and carry 1 carry 1 and again this it will be added with this, so 0 plus 1 this will be 1 sum is 1 and carry 1, so sum is 1 carry 1.

Now, again this is 0 plus 1 1 and 1 plus 1 sum 0 carry 1, now if it is 3 1 that similar this is similar 3 1 to be added this will be sum 1 carry 1 and that should be come here. So, 1 plus 1 0 carry 1, again this is 1 0 plus 0 plus 1 1 1 plus 1 0 carry 1 1 plus 1 plus 1 1 carry 1 1 plus 1 0 carry 1 1 plus 1 plus 1 carry 1 sum 1. So, this is my binary addition and these are the carries, these are my carries and these two operands.

(Refer Slide Time: 45:19)



Now, if I know binary addition then; obviously, the same rule I can apply to binary subtraction. We see the binary subtraction. Now, see that again if I write the addition rule they are the rules are 0 plus 0 equal to 0 0 plus 1 equal to 1 1 plus 0 equal to 1 1 plus 1 equal to 0 carry 1. So, here carry if you write 0 0 0 carry 1, this my sum, the same rule now see here I am subtracting 1 0 1 1 1 from the value 1 0 0 1 1 0 1.

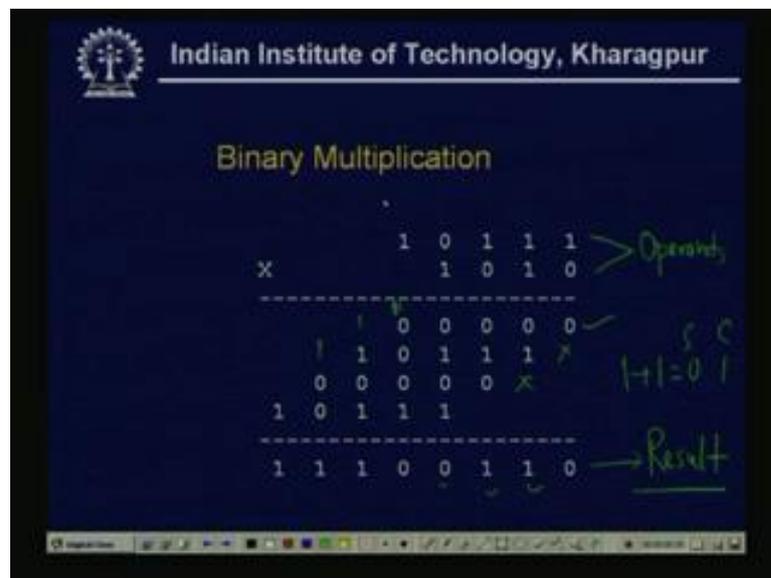
So, first if it is see that 1 minus 1; that means, see here I can take from here as if these 1 is see these 1 is subtracted from these; that means 1 plus 0 equal to 1. So, this if I subtract this thing, so 1 minus 1 equal to 0, so the same thing I have done 1 minus 1 equal to 0. Now 0 minus 1, what do we mean by 0 minus 1, so; that means, I have to take these 1, so 0 minus 1 that will be 1, so the same thing I have done 0 minus 1 it is 1.

So, from our normal subtraction rule or the decimal system that what we know, actually I borrow 1, because 0 is lesser than 1, so I borrow a 1 and that is 1 0. So, these are my this is my borrow for this subtraction bit. So, this is 1 is borrowed, so 1 0 this is 1. Now,. So, this becomes this becomes this becomes 0, because I have borrowed. So, this is less this become 0.

So, now again 1 I have to subtract from 0, so again I it should be borrowed and this is 1 0. So, again it is 1 and it is again borrowed, so this becomes 0. Now see that 0 and 0, this 0 minus 0 is 0, so actually here it is not it is not borrow 0 minus 0, it is 0 there is no borrow.

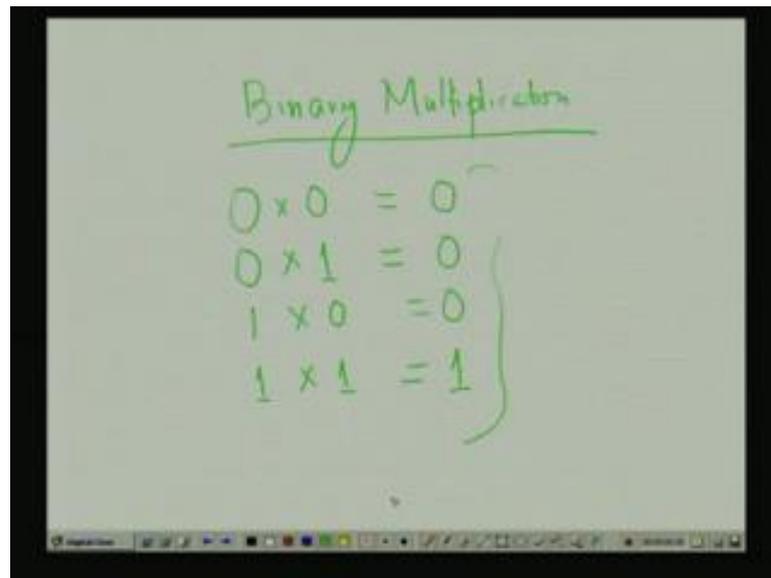
Now, again 0 minus 1 0 minus 1, again this is borrow, see 1 is 1, I want to subtract from 0. So, this should be that 1 subtracted from 0, this should be 1, this should be again this is actually 1 0. So, this is 1 again it is borrowed, this becomes 1 0 similarly this is 1 and again this becomes this 1, this is this become 0, 0 1 as 0 0, this is my result. So, this is the binary subtraction. So, actually there, what we have seen the binary rule or binary addition has been applied.

(Refer Slide Time: 50:12)



Now, binary multiplication now binary multiplication means the same to our decimal multiplication again the only the multiplication rule is different; that means, here again there are only digit numbers 0 and 1.

(Refer Slide Time: 50:36)



So, what we can say that what were the basic rules, see for binary multiplication again this can be 0 into 0 it is defined as 0. Then, again 0 into 1 then again it is 0 then 1 into 0; obviously, that will be 0 and then this is 1 into 1 again this is a 1. So, this only this multiplication is different and addition will be just now the binary addition we have defined that we have to do.

Now, if I apply, say I have two such operands say I want to multiply 1 0 1 1 1 and 1 0 1 1. So, the 0 multiplied by anything already we have seen that will be 0, then again I am giving the cross this will be multiplied by 1. So, 1 multiplied by 1 into 1 1 and 1 into 0 0, so the same thing again that it will be multiplied by 0, then it will be multiplied by 1 now only binary addition I have to apply.

So, this is 0 now 0 plus 1 already we have seen 0 plus 1 is 1, 0 plus 1 1 and 1 plus 0, so this is 1. Now, 0 plus 1 1 plus 0 1 1 plus 1 sum 0, because we know 1 plus 1 is sum 0, but carry 1, this becomes now sum 0 carry 1 carry 1 say I am giving 1 carry 1. Now, if I add again 3 0 means 0 and 1 plus 1 0 carry 1. Now, again 1 plus 1 0 0 plus 1 1, but carry 1 again 0 plus 0 plus 1 1 and this 1 1, so this is my result. The similar the binary addition rule is applied and only the multiplication is slightly different otherwise that total multiplication rule is same as that of our decimal notation.

(Refer Slide Time: 53:52)

Indian Institute of Technology, Kharagpur

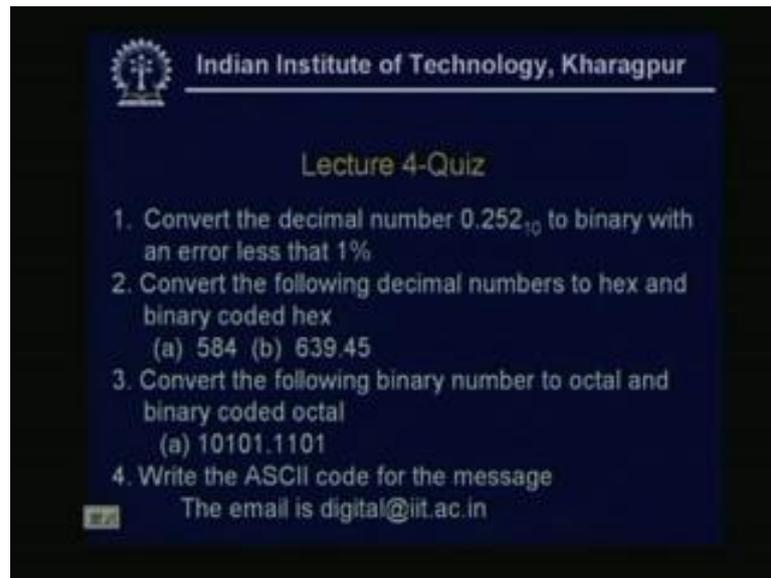
Lecture 4-Quiz

1. Convert the decimal number 0.252_{10} to binary with an error less than 1%
2. Convert the following decimal numbers to hex and binary coded hex
(a) 584 (b) 639.45
3. Convert the following binary number to octal and binary coded octal
(a) 10101.1101
4. Write the ASCII code for the message
The email is digital@iit.ac.in

Now, we see that our today's quiz question, that lecture 4 quiz the first thing is the convert the decimal number 0.252 in decimal base to binary with an error less than 1 percent. Second is, convert the following decimal numbers to hex and binary coded hex, so that this is the two decimal numbers are given one is pure integer. One has the integer as well as non-integer portion and we have to convert that thing to hexadecimal numbers as well as BCH the binary coded hexadecimal.

Now, convert the following binary numbers to octal and binary coded octal and this is very similar to the question number 2, but only the binary is itself given. And, write the ASCII code for the message the email is digital at IIT dot ac dot in. So that means, I want that this, what will be the ASCII code of these message, if when I type this when I type this message that the email is digital at the rate IIT dot ac dot in. Then, how it is converted to that ASCII, so to do that thing so, these are the four questions for lecture 4 quiz. So, today we end at the at this point.

(Refer Slide Time: 56:11)



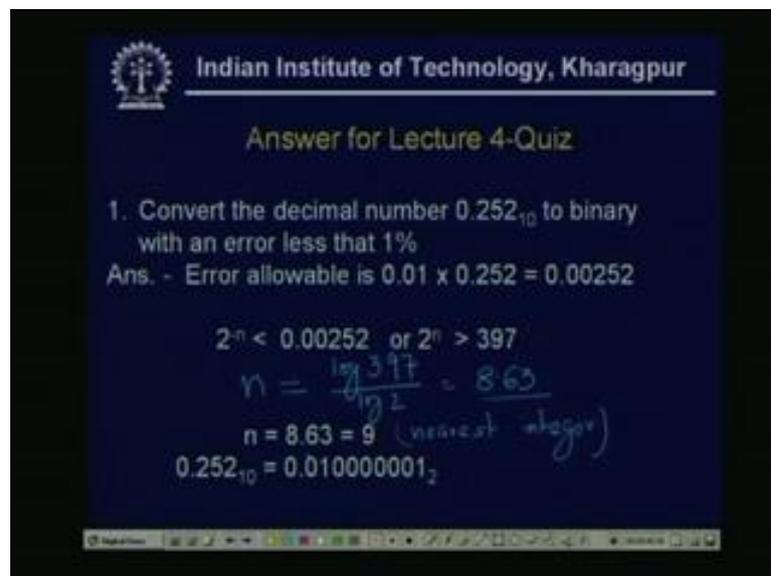
Indian Institute of Technology, Kharagpur

Lecture 4-Quiz

1. Convert the decimal number 0.252_{10} to binary with an error less than 1%
2. Convert the following decimal numbers to hex and binary coded hex
(a) 584 (b) 639.45
3. Convert the following binary number to octal and binary coded octal
(a) 10101.1101
4. Write the ASCII code for the message
The email is digital@iit.ac.in

Preview of next lecture before that we quickly go through the answers of the last day's lecture quiz.

(Refer Slide Time: 56:25)



Indian Institute of Technology, Kharagpur

Answer for Lecture 4-Quiz

1. Convert the decimal number 0.252_{10} to binary with an error less than 1%

Ans. - Error allowable is $0.01 \times 0.252 = 0.00252$

$$2^{-n} < 0.00252 \quad \text{or} \quad 2^n > 397$$
$$n = \frac{\log 397}{\log 2} = 8.63$$

$n = 8.63 = 9$ (nearest integer)

$$0.252_{10} = 0.010000001_2$$

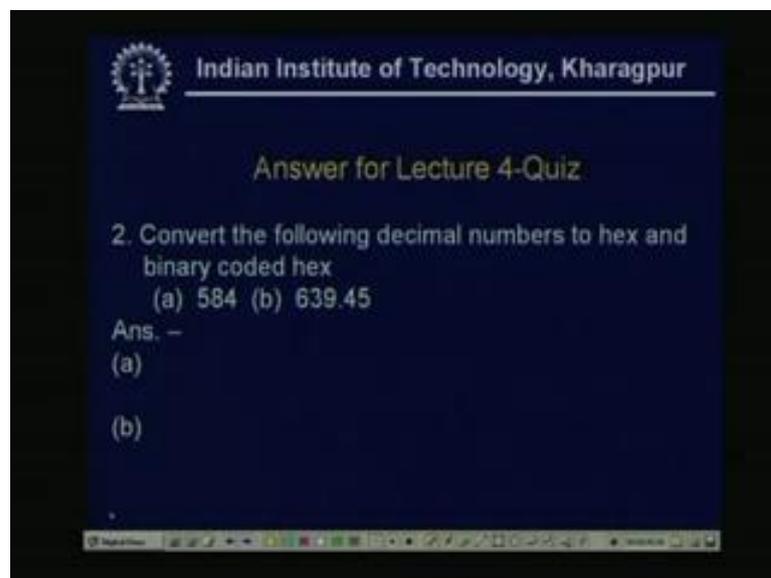
So, last day's quiz there are 4 questions given and we see the answers. The first question was to convert the decimal number 0.252 decimal to binary with an error less than 1 percent. Now, as the error is 1 percent, so the, what will be the error on 0.252, so we will be multiplying this thing by 0.01 into the number 0.252. So, this is equal to 0.00252.

Now, the truncation error; that means, when we convert the decimal number to the binary up to, what digit we will convert it, so if it is up to n bits; that means, after decimal point. If we take the value up to n bits then we know the error is 2 to the power minus n last lecture we discuss that thing, so these error should be less than these 1 percent error means 0.00252 . So, 2 to the power minus 1 is less than 0.00252 that can be written as 2 to the power n greater than 397 .

So from this expression, we can get the n if I take the log on both side, then it will be log of 397 divided by log 2 and this is value equal to 8.63 . Now, as n is the number of bits, so this should be some integer value and that is, why we are, taking the nearest integer is 9 that is my nearest integer.

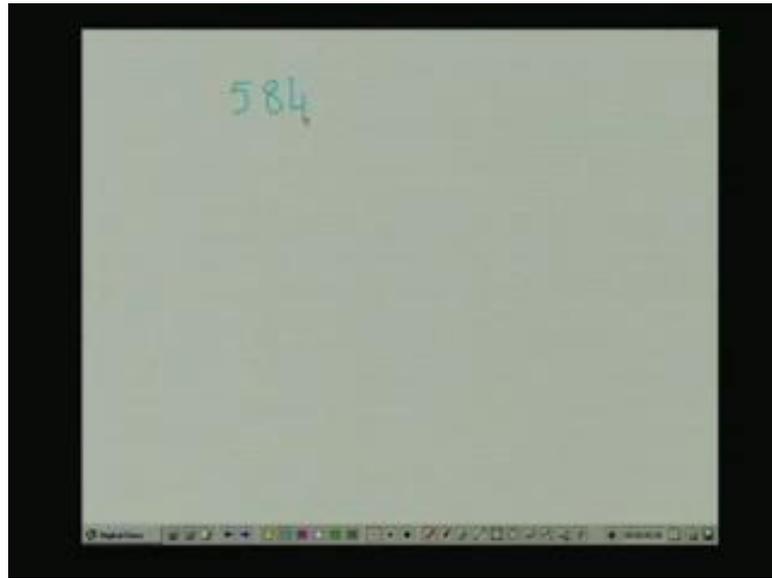
So, when we convert 252 as binary then we have to if we take the number of bits after decimal point is not n means in control 9 ; that means, if we take 9 bits then it will be the error will be less than 1 percent. So; that means, point 252 in decimal is equal to 0.01000001 in binary; that means nine bits after this decimal point.

(Refer Slide Time: 58:55)



Now, you see the next question, next 1 is the convert the following decimal numbers to hex and binary coded hex.

(Refer Slide Time: 59:05)



The first 1 we see that 584 is the number, now, 5 8 4 if I...