Switching Circuits and Logic Design

Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

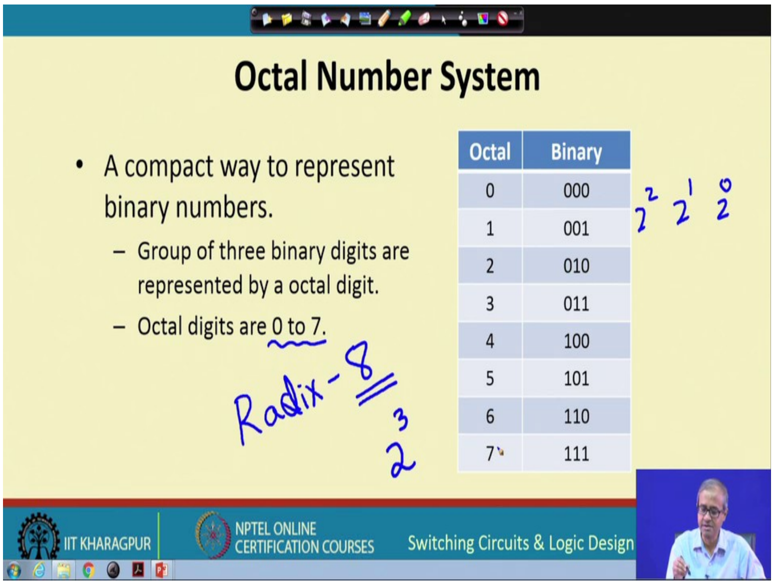
Lecture - 02
Octal and Hexadecimal Number Systems

Welcome back. In this lecture we shall be mainly talking about something called Octal and Hexadecimal Number Systems. But first let us try to understand the motivation why do we need this. In the last lecture if you recall, we talked about the decimal number system which we are all familiar with. Since our childhood we have been taught decimal numbers. So, all arithmetic that you do by hand are based on decimal number system the things that have been taught in our school. Binary number system is something which all computer systems are based on, that is why we also have to learn binary number system.

Now, one problem with binary number system as you can understand that is that, because the weights of the digits are in powers of two, the number of digits you require to represent a number may require many digits or bits. Take an example. So, there is one example which you took in the last lecture, a number 64. So, in decimal to represent 64 we need only 2 digits right. But in binary we need one followed by 6 zeros; that means, 7 bit. So, when we express a number or when you write down a number, we may need a large number of bits. So, it may be inconvenient to write so many bits. So, this octal and hexadecimal numbers are in a sense a compact way of representing binary numbers that is one way you can look at it.

Of course, they are separate number systems in their own right, but their main use is to represent binary numbers in a compact way ok. Let us start with this motivation.

(Refer Slide Time: 02:23)



The octal number system: octal number system this octal means 8. Basically this is a weighted number system with the radix of 8. So, you can say here our radix is 8, which means our digits are 0 to 7 let us say. And in this table because I said that one of its use is to represent binary numbers in a compact way. Here one thing you note that the radix that we chosen here as 8, which is some power of 2 this is important.

$$Radix\, 8 = 2^3$$

Because it is power of 2, so the idea is that every 3 bit binary number can be represented in octal as a single digit like you look at binary number 0 0 0 what is the value in decimal? 2 to the power 0 (2^0), 2 to the power 1 (2^1), 2 to the power 2 (2^2) this will be the weights 2 to the power 0, 2 to the power 1 and 2 to the power 2 multiply them this is 0. So, this corresponds to 0. 0 0 1, 0 0 1 the last one is 1, it corresponds to 1. 0 1 0, second one is 2, it is 2. 011 second and third 2 and 1, 3; 1 0 0 this is 4; 1 0 1, 4 and 1, 5; 1 1 0, 4 plus 2, 6; 1 1 1, all 3 are there 7. So, here you see this octal digit 0 to 7 is each of them they correspond to a 3 bit binary number. This is one correspondence that you have to look at here.

$$(000)_2 = 0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 = (0)_8$$
$$(001)_2 = 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = (1)_8$$
$$(010)_2 = 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = (2)_8$$
$$(011)_2 = 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 2 + 1 = (3)_8$$

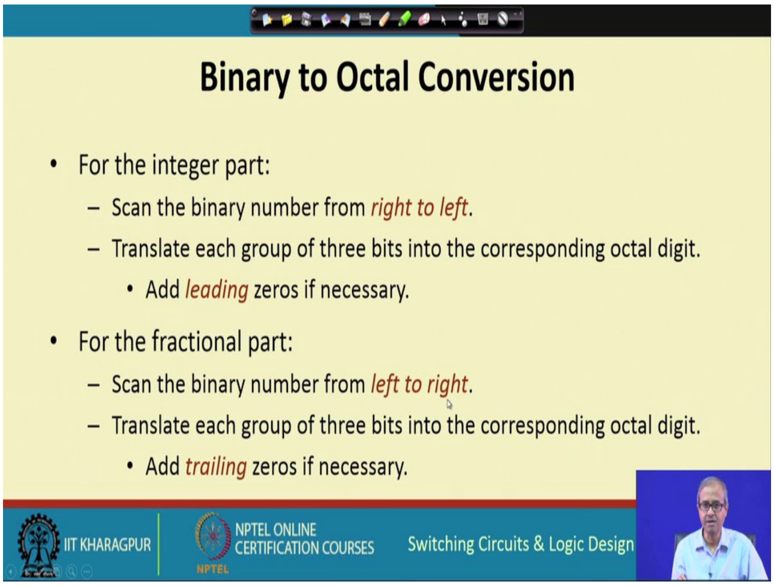
$$(100)_2 = 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 = (4)_8$$

$$(101)_2 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 4 + 1 = (5)_8$$

$$(110)_2 = 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 4 + 2 = (6)_8$$

$$(111)_2 = 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = (7)_8$$

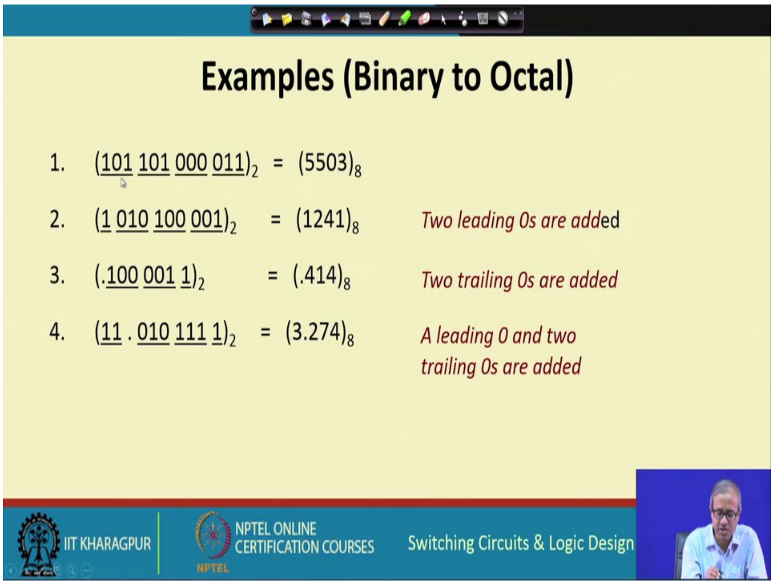
(Refer Slide Time: 04:27)



Now when you talk about binary to octal conversion, this is basically following this principle. So, when you have an integer part, you scan the number from right to left. I will take examples. Given a binary number you scan the number from the least significant position right to the most significant digit position left and during scanning, you make groups of 3 bits and each group of 3 bits will be replaced by the corresponding octal digit.

This is the basic rule that we are following. You scan the binary number from right to left, group 3 digits each and each group of 3 digit you replace by the corresponding octal digit. Only for the most significant part, if the number of bits left is less than 3 you can pad zeros in the beginning to make it 3. So, add leading zeros if necessary. For the fractional part you do the same, but now the scanning is from left to right why? Because for a fractional part you can add zeros at the end, you cannot add zeros in the beginning because the value will change. But for the integer part you can add zeros in the beginning that is the only difference right. So, fractional part scan it from left to right, do the same thing make groups of 3 bits and for the last part you can add some trailing zeros is required. Let us take some examples.

(Refer Slide Time: 06:06)



Take a binary number like this 101 101 000 011. So, the numbers of bits are divisible by 3 there are 12 bits. So, you make groups of 3 3 3 starting from least significant side 3 3 3 3. 011 means 3, 000 means 0, and this 101 means 5, and 101 means 5. So, it is a straight forward conversion. Now you can see the advantage of octal. I said octal is a way to represent a binary number in a compact way. This is why I said that. There is a one to one correspondence, for conversion you do not have to carry out any multiplication or division like in decimal to binary or binary to decimal.

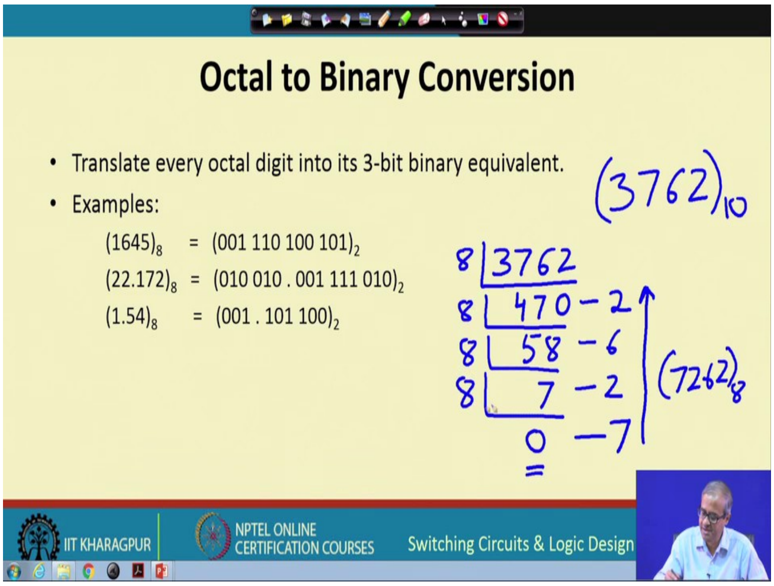
$$(101101000011)_2 = (\underline{101}\,\underline{101}\,\underline{000}\,\underline{011})_2 = (5\,5\,0\,3)_8$$

Let us take another example where there are you can count 10 digits. So, again scan from right to left 100 is 1, this is 4, 010 is 2 and you have a single one left. You add 2 zeros in the beginning it becomes 001, which is 1. This is octal, this 8 indicates octal. So, for this case 2 leading zeros are added. Take a number means a pure fraction. So, pure fraction is scan from left to right. So, 100 is 4, 001 is 1, then you have a single one. So, you have to add 2 zeros it makes 100 which is 4 and for a mix number both integer and fractional part, for integer part we scan from right to left it is 11 add one 0 it is 3. Here 010 is 2, 111 is 7, 1 add two zeros it becomes 4.

$$(1010100001)_2 = (\underline{001}\,\underline{010}\,\underline{100}\,\underline{001})_2 = (1\,2\,4\,1)_8$$
$$(.1000011)_2 = (.\,\underline{100}\,\underline{001}\,\underline{100})_2 = (.4\,1\,4)_8$$
$$(11.0101111)_2 = (\underline{011}.\underline{010}\,\underline{111}\,\underline{100})_2 = (3.2\,7\,4)_8$$

So, a leading 0 is added here in the integer part, and 2 trailing zeros are added in the fraction part right. So, binary to octal is done like this.

(Refer Slide Time: 08:20)



Now octal to binary is even simpler. Just you take an octal number and each octal digit is replaced by its 3 bit binary equivalent like 1645 replace 1 by 001, replace 6 by 110, 4 by 100, 5 by 101. Similarly a fractional number 2 by 010, 2 010 again then dot, 1 001, 7 111, 2 010 and so on.

$$(1645)_8 = (\underline{001}\,\underline{110}\,\underline{100}\,\underline{101})_2$$
$$(22.172)_8 = (\underline{010}\,\underline{010}\,.\,\underline{001}\,\underline{111}\,\underline{010})_2$$

So, you see converting a binary number to octal or an octal number to binary is very trivial, but if you want to convert for some reason, decimal to octal or octal to decimal you can follow the same rule like in binary. Like I am giving one specific example for decimal to octal conversion. Let us take a specific example. Let us say I have a number 3762, this is a decimal number. So, I want to convert it to octal. So, what I do, I take this number, I repeatedly divide by 8, I divided by 8. 8 fours 32, 5, 7 56, 0 with a remainder of 2, divide by 8 again.

8 5 47, 8 8 64 and remainder of 6. Divide by 8 again, 8 7 56, remainder of 2, divide again 0 with a remainder of 7. You have you have arrived at 0. So, stop and you take it in the reverse order 7 2 6 2. So, 7262 is the equivalent representation in octal. So, conversion from decimal

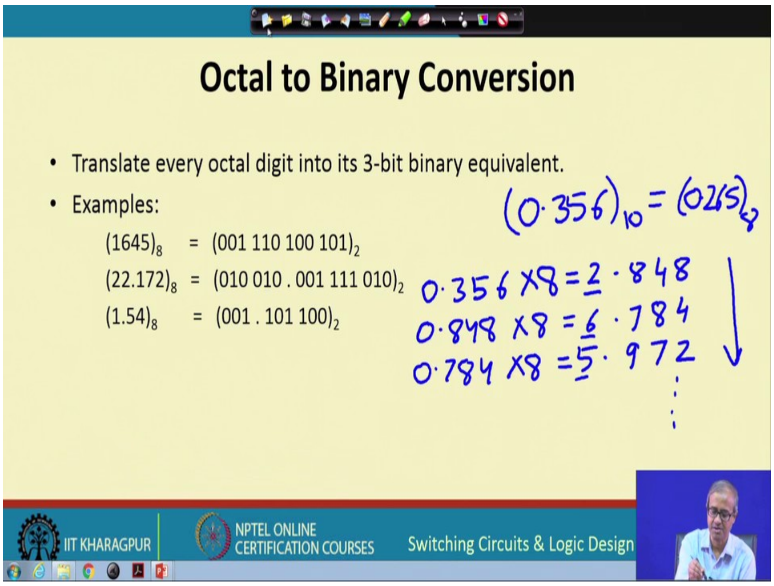
to octal can be done in this way following a principal, which is almost identical to that for decimal to binary only difference is here you are dividing by 8 instead of dividing by 2. For fractional part it is same you this time instead multiplying by 2 you will be multiplying by 8 let me just take an example again.

¿
8 | 3762 ¿¿

8 | 470 − 2 ¿¿

¿
8 | 58 − 6 ¿¿
¿
8 | 7 − 2 ¿¿
−7
8 | 0 ¿¿

(Refer Slide Time: 11:04)



Let us say I have a decimal number let 0.356 this is decimal. I want to convert it to octal. So, what I do? 0.356 I multiply by 8. So, how much is the? 4, 4, 24, 28 and integer part is 2. So, you remember this integer part. Now fractional part is 8 4 8. So, you take 848 multiplied by 8. So, now, it become 6, 38, 3, 7 and 6. So, now, the integer part 6, take 784 multiply by 8, 3, 6 like this.

And this continues. So, now, if you take it in the same order 265, so this will be equivalent to 0.265 in octal right. This is how we can convert from directly from decimal to octal or octal to decimal.

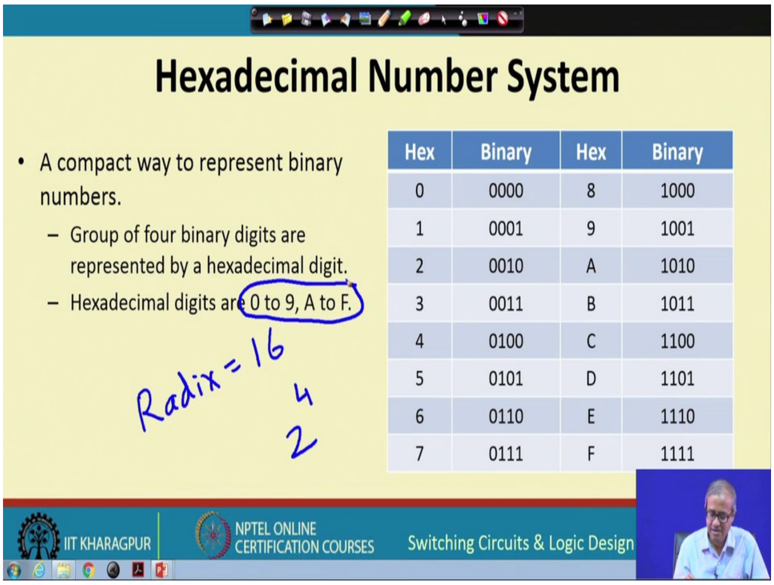
$$0.356 \times 8 = \underline{2}.848$$

$$(0.356)_{10} \cong (0.266)_8$$

$0.848 \times 8 = \underline{6}.784$

$0.784 \times 8 = \underline{6}.272$

(Refer Slide Time: 12:30)



Now, let us come to hexadecimal, which is one step further. Octal is 8, hexadecimal is 16. So, the earlier case we are grouping 3 binary digits 3 bits to form one octal digit, now we will be grouping 4 binary digits or bits to form a hexadecimal digit. So, hexadecimal will be even more compact in terms number of digits.

$$Radix\, 16 = 2^4$$

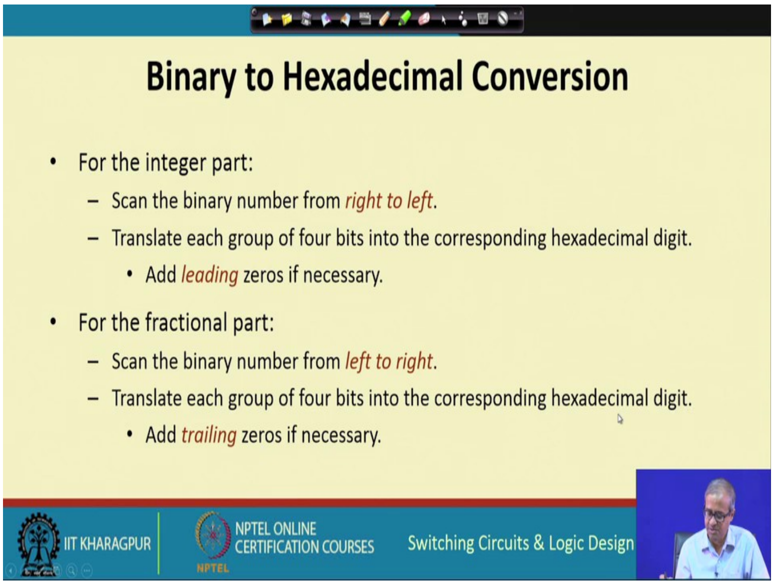
So, in case of hexadecimal number system, radix is defined to be 16. So, 16 is again a power of 2 that is why binary to hexadecimal conversion are easy and this 16 digits are defined as follows: The first 10 are 0 to 9 last 6 are A to F, A B C D E F. So, 0 1 2 3 4 up to 9, then A B C D E F. This is how the 16 digits are defined in hexadecimal. So, this table shows the different hexadecimal digits 0 up to F and the corresponding binary equivalents. See you take any one 0110 means what? You multiply by the weights it is 6 decimal. So, it is equivalent to 6 digit.

$$(0110)_2 = 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = (6)_{16}$$

Similarly, you take let us say 1100 it is 8 plus 2, 12. C means 12. See 9, 10, 11, 12 13, 14, 15. So, it represents the digits. Similarly 1111 means 8 4 2 1, 15; 15 means F; so every 4 bit combination is equivalent to one binary or is equal to one hexadecimal digit, this is the basic idea.

$$(1100)_2 = 1\times 2^3 + 1\times 2^2 + 0\times 2^1 + 0\times 2^0 = 8+4 = 12 = (C)_{16}$$
$$(1111)_2 = 1\times 2^3 + 1\times 2^2 + 1\times 2^1 + 1\times 2^0 = 8+4+2+1 = 15 = (F)_{16}$$

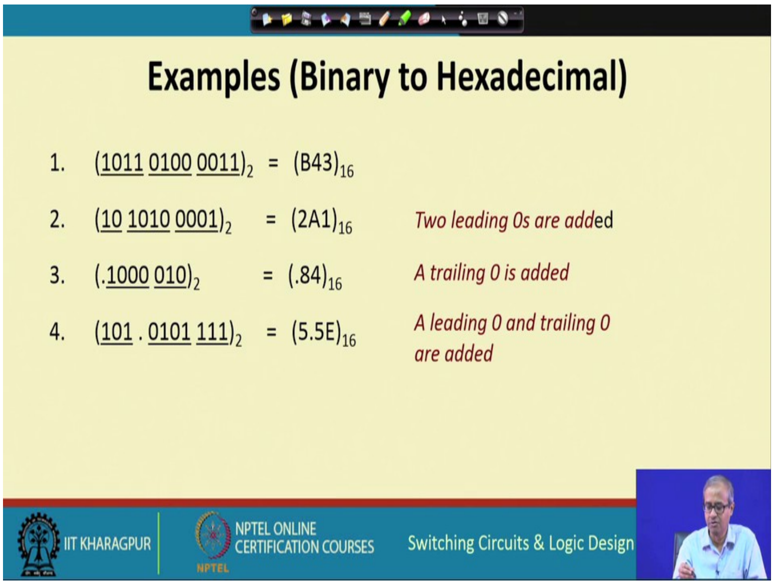
(Refer Slide Time: 14:38)



So, when you talk about binary to hexadecimal conversion it is quite similar to octal. For the integer part you scan it from left to right, right to left sorry binary number is scanned from right to left and instead of 3 bits now you group 4 bits. And each group of 4 bits you replace by the corresponding hexadecimal digit and the last one if it is less than 4, you add leading zeros.

And similarly for fractional part, you scan from left to right. So, again you make groups of 4 replace it by the hexadecimal digits and add trailing zeros if required.

(Refer Slide Time: 15:21)



Take some examples here, this is a binary number 1011 is what? 8 2 1; that means 11, 11 means is the digit B, 0100 is 4, 0011 is 3 this is the hexadecimal equivalent 16 indicates hex. Similarly this number you scan from right to left 0001 is 1, 1010 is A, it is 10, 10 means A, 1 0 at 2 zeros 0010 is 2. For fractional numbers scan from left to right 1000 is 8, 010 add one 0, 0100 which is 4.

Similarly, another example here, here you add a 0, 0101 is 5, this is 5 and 111 and a 0, it is 14 which is E ok. So, this binary to hexadecimal is fairly simple almost same as binary to octal only the size of these groups are different.

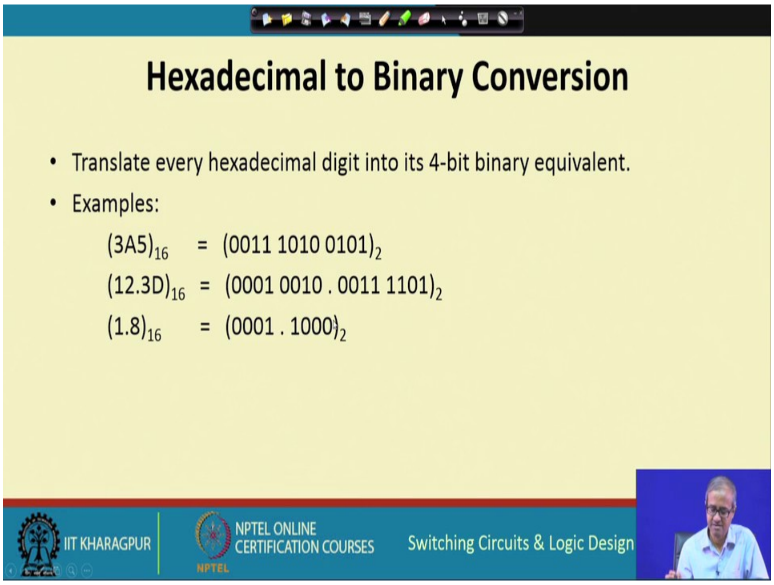
$$(101101000011)_2 = (\underline{1011}\ \underline{0100}\,\underline{0011})_2 = (B43)_{16}$$

$$(1010100001)_2 = (\underline{0010}\,\underline{1010}\,\underline{0001})_2 = (2A1)_{16}$$

$$(.1000010)_2 = (.\,\underline{1000}\,\underline{0100})_2 = (.84)_{16}$$

$$(101.0101111)_2 = (\underline{0101}\,.\,\underline{0101}\,\underline{1110})_2 = (5.5E)_{16}$$

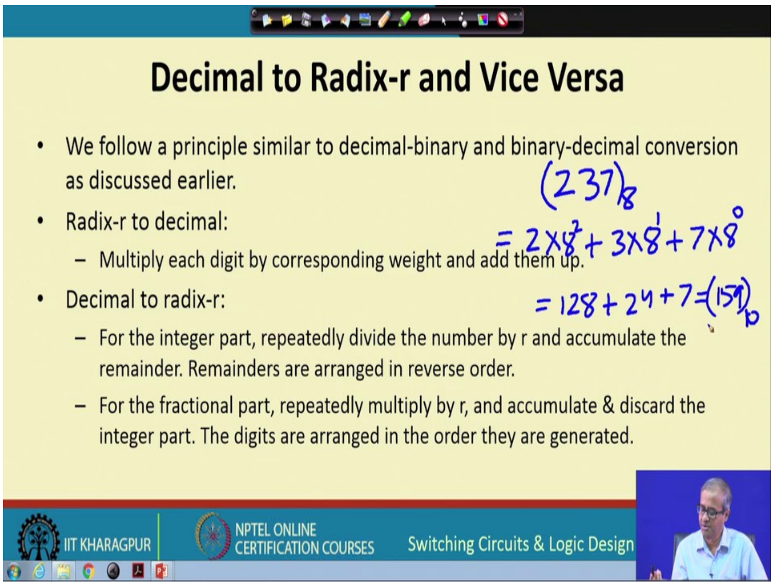
(Refer Slide Time: 16:30)



Hexadecimal to binary again is very simple, you take the hexadecimal digit replace them by their 4 bit binary equivalents. 3 A 5 this is 3, this is A this is 5. 12.3 D this is 1 2.3 and D. 1.8 18 ok.

$$(3A5)_{16} = (\underline{0011}\,\underline{1010}\,\underline{0101})_2$$
$$(12.3D)_{16} = (\underline{0001}\,\underline{0010}.\underline{0011}\,\underline{1101})_2$$
$$(1.8)_{16} = (\underline{0001}.\underline{1000})_2$$

(Refer Slide Time: 16:54)



So in general I just give an example just sometime back for r equal to 8.

So, if you have any arbitrary radix r number and if you want to convert from decimal to radix r and vice versa, you follow a similar principle as we showed for binary. For radix r to decimal you multiply each digit by corresponding weight and add them up. Like I am giving an example, let us say I talk about an octal number. Let us say I have an octal number 237 and I want to find out its value, I want to convert it to decimal. So, each digit position will be some have they will have some weights. So, it be 2 multiplied by 8 to the power 2 plus 3 multiplied by 8 to the power 1 plus 7 multiplied by 8 to the power 0. So, 8 square is 64, 64 into 2 is 128 plus 24 plus 7 so, it comes to 159.

$$(237)_8 = 2 \times 8^2 + 3 \times 8^1 + 7 \times 8^0 = (159)_2$$

This will be the equivalent decimal number right. So, radix r to decimal will follow this principle, multiply each digit by the weight, add them up. Decimal to radix will be the this will using repeatedly division or repeated multiplication for the integer part you repeatedly divide by r and take the remainder, and consider reminders reverse order.

I illustrated this for octal r equal to 8 and similarly for the fractional part, you multiply by r and accumulate the integer part. This also I had shown for octal. So, in this lecture we have basically looked at the octal and the hexadecimal number system, we saw how we can convert binary to hexadecimal and octal and vice versa because I repeatedly said, binary numbers are most important to us when we are talking about designing digital circuits and computer design, because they all work on the principle of the binary number system. And also I talked about for a general radix r, how we can convert a decimal number to radix r or vice versa. These basic techniques will help you later when you want to convert any arbitrary number from one radix system to another.

Thank you.