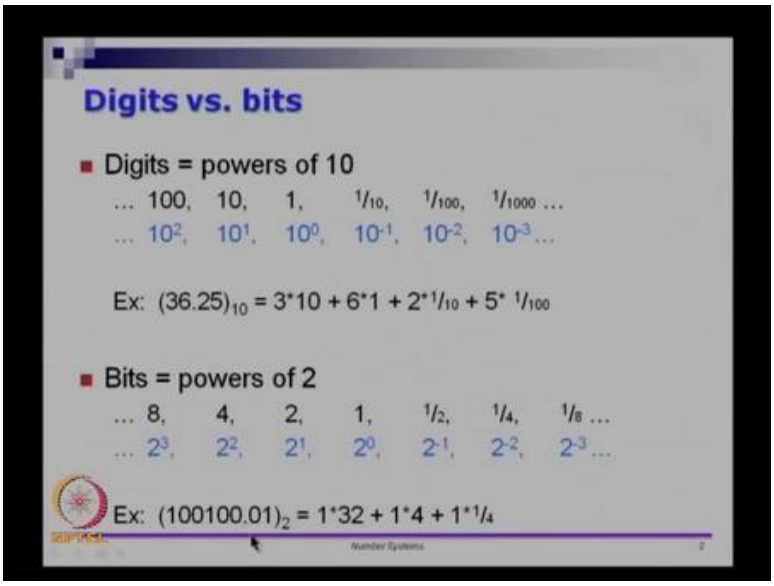
Digital Circuits and Systems
Prof. Shankar Balachandran
Department of Electrical Engineering
Indian Institute of Technology, Bombay
And
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Module - 12
Number Systems

Hi, welcome to the last module for this week. In this module, we will go, we will learn the basics of number systems. So, I hinted up on this in the previous modules about number systems. So, we will go in little more detail about number systems in this module. So, we already saw this notion of digits versus bits.

(Refer Slide Time: 00:47)



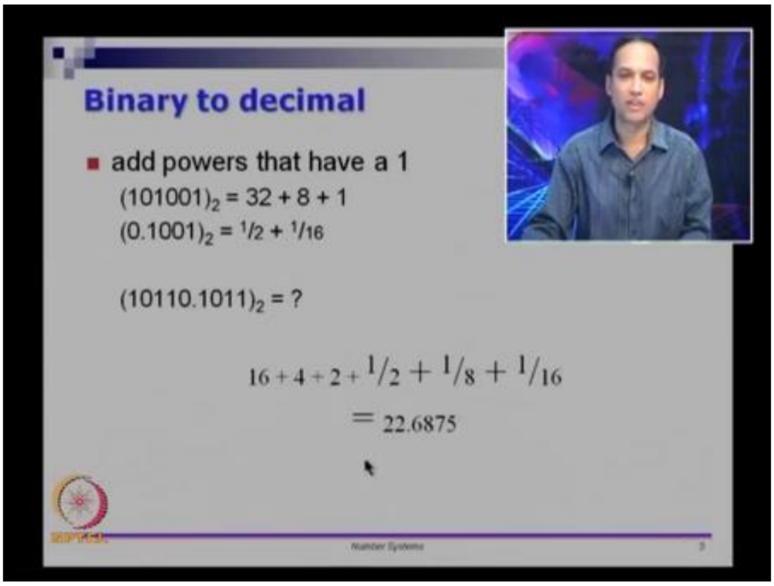
So, when we say digits, the digits are in the powers of 10. So, we know that when we have 1s, 10s, 100s and so on. They are actually 10 power 0, 10 power 1, 10 power 2 and so on. And we also have 1 by 10th, 1 100th and 1000th and so on, that is as good as 10 power minus 1, 10 power minus 2, 10 power minus 3 and what not. So, if you go and look at 36.25 in decimal, then we can assign the weights. So, you start at the decimal point, go to the left side and move to the left there to start with. So, you start with the units here and tense here. So, that is unit and this is 10 power 0 and that is 10 power 1 and, when you, and then you move to the right side you take 2 into 1 by 10 plus 5 into 1 by 100. So, this is a proper positional weight system.

So, in this positional weight the decimal decides what weights you get. Everything to the left of the decimal gets increasing powers of 10s starting from 0 and everything to the right of the decimal gets decreasing powers starting from 10 power minus 1. So, that is a positional weight system and this automatically extends to binary also. So, if in binary what you do is, if you see a dot given in a binary number you start from the left side assign from weight 2 power 0, 2 power 1 and so on to the left and start with 2 power minus 1, 2 power minus 2 and so on, to the right. So, if we go and look at it from one side it is actually only increasing from left to right.

So, for example, if I even if I take 36.25, this is 10 power minus 2, this is 10 power minus 1, this is 10 power 0 and this is 10 power 1, right. So, from the rightmost thing, it, if we move to the left side, it only increases. But when, when you look at it from the decimal point of view where we have the dot, when we go on to the right side, it is decreasing and when we go in to the left side, it is increasing, right. So, the same thing applies here. If you go and look at this, the binary number here. So, let us see what is the value of this one.

So, you start from here. So, this is 0, this is 2 into 0, this is 4 into 1, this is 8 into 0, 16 into 0 and 32 into 1. So, we have 32 plus 4, so this is 36. And on the right side this is 0 into 2 power minus 1 plus 1 into 2 power minus 2. So, 2 power minus 2 is 4. So, that is 1 into 1-4th, that is the value here, 1 into 32 plus 1 into 4 plus 1 into 1 by 4. In fact, this is the binary representation for 36.25, right. So, you have 36.25 here. This is the actual equivalent binary representation only that for 36, you have representation on the left side, for 0.25 you have representation on the right side of the decimal. The decimal point appears exactly at the same point as in the integer value is to the left of this dot and the decimal value or the fraction is to the right of the dot. The same thing happens in binary also, the fraction is in the right side and the actual integer to represent the number is on the left side of dot.

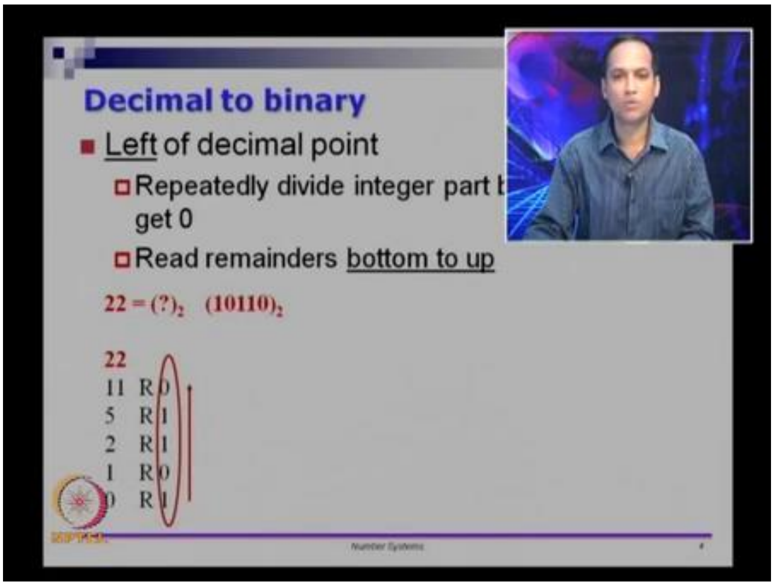
(Refer Slide Time: 03:51)



So, how do we convert binary to decimal? So, we, I already talked about this. You take the number and then you assign weights to it. So, you always start from the right side, you start giving, increasing powers of 2, starting with 2 power 0. So, this is 2 power 0 into 1 plus 2 power 3 into 1 plus 2 power 5 into 1, that is same as 41. Then, for this example, this is 1 into 2 power minus 1 plus 1 into 2 power minus 4, that is the same as 1, one-half plus 1-16th. So, let us not worry about the decimal equivalent, but you, I know you can do the fractions here. So, it is one-half plus 1-16th.

So, let us see what the equivalent for this one is, 10110.1011 is. So, if you start from here, so this is 2 power 0, this is 2 power 1. So, we have the two term here, then this is 2 square. So, we have 4, 2 power 3 will be missing, and then you will have 2 power 4. So, the first terms, so the first three terms are the integer representation, they are 16 plus 4 plus 2, which is 22. And if you look at the right terms, this is 2 power minus 1 should appear here, then 2 power minus 2 should not be there, then we have 2 power minus 3 and 2 power minus 4. So, that is one-half plus 1-8th plus 16th that is the same as 0.6875. So, the left side gives us the integer 22, the right side gives us a 0.6875. Overall, we have 22.6875. So, this is binary to decimal.

(Refer Slide Time: 05:24)

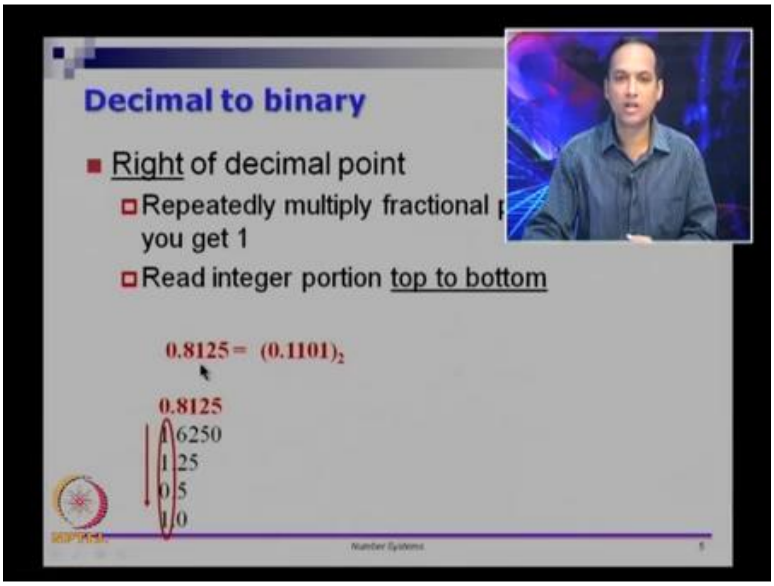


Decimal to binary is slightly involved, you might have learnt this in school, but I want to give it here, so that it is for the sake of completion. So, what you have to do is, for the left of the decimal point you repeatedly divide the integer, integer by 2 until you get 0 and read the remainders from bottom to top. So, let us see this.

If I want to find out what 22 is represented in terms of binary, then I will take 22, divide that by 2, so it gives me 11 as the quotient and remainder is 0. You divide 11 by 2, that gives me 5 as the quotient and remainder as 1. Then, take 5 and divide by 2, you get 2 as a quotient and remainder 1. And finally, you get 2 divided by 1 and that gives as remainder 0 and quotient 1 and you take that 1 and divide by 0, you get remainder, you get quotient 0 and remainder 1. What you have to do is, you have to start with the number and keep dividing until you get 0.

So, the remainder maybe 0 or 1, but you have to stop and the moment you get quotient equals 0 once, you stop. The rule to read this is, read from bottom to up. So, if you read from bottom to up, we are going to take only the remainders, we discard this, we do not need the quotient, we look at only the remainders and we will go bottom to top. And if you read this here, it is 10110. So, that is the input. So, it is, 22 is 10110 in base 2.

(Refer Slide Time: 06:57)

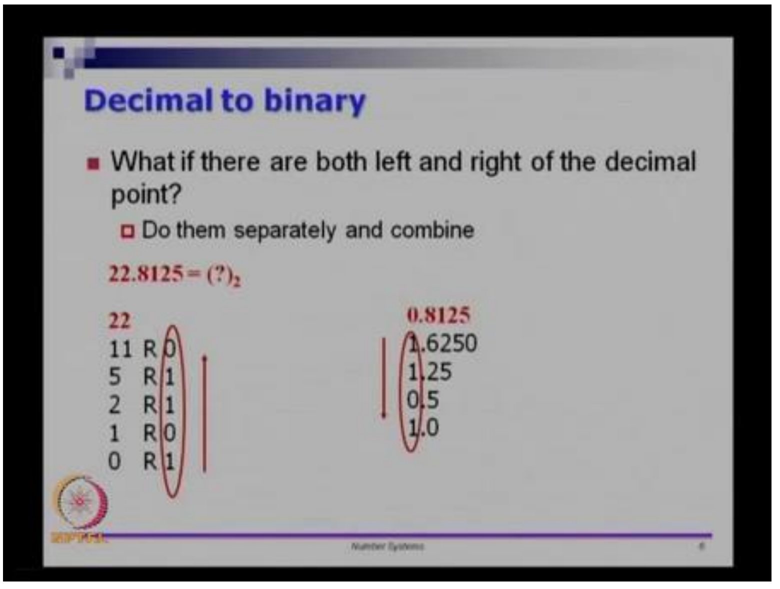


Let us see what happens to the right of the decimal point. So, the right of the decimal point we are going to do something slightly different. So, for the left side we repeatedly divided, so the right side we are going to repeatedly multiply until we get a 1. So, the left side you repeatedly divide until you get a 0 as quotient and in the right side you repeatedly multiply by 2 until you get 1. Let us see what it is. So, let us see this example, 0.8125, I want to find out what is binary representation for it.

I take 0.8125 and multiply it by 2, this you represent this in decimal now, but it gets you 1.625. So, you keep the 1 aside, take just 0.625 and multiply that by 2. So, 0.625 into 2 is 1.25. So, remember it is, this 1 is not 1.625 into 2, we discard, we keep the 1 aside, take only this 0.625 and multiply that into 2, that is, 1.25. Then, again keep this 1 aside, will come to it later, take only the decimal part, which is 0.25 and multiply by 2, that is, 0.5. Again, keep the 0 aside, take 0.5 and multiply by 2, it gives me 1.0. At this point we have it, 1.0. This is when we stop.

The moment you hit 1.0, you should not have anything on the right side and we get 1, number 1, that is when we stop. At this point what we do is, we go and note down all the, all the things that we got to the left side of the decimal point. So, one after the other we were keeping this aside, you keep this and read it from top to bottom. So, you write that down, 0.8125 is 0.1101 base 2. So, you can verify that quickly. So, this one is going to get a power of half. So, this is 1 by 2, which is 0.5 plus 1 by 4, right, because of this one you get a 0.1 by 4. So, 0.5 plus 0.25 is 0.75. Then, 1 by 8 is missing and you have 1 by 16th. So, 1 by 16th is 0.0625. So, 0.75 plus 0.0625 is 0.8125. So, if you are not sure go and verify it more carefully, you put the weights and check it more carefully, you will see that there.

(Refer Slide Time: 09:25)

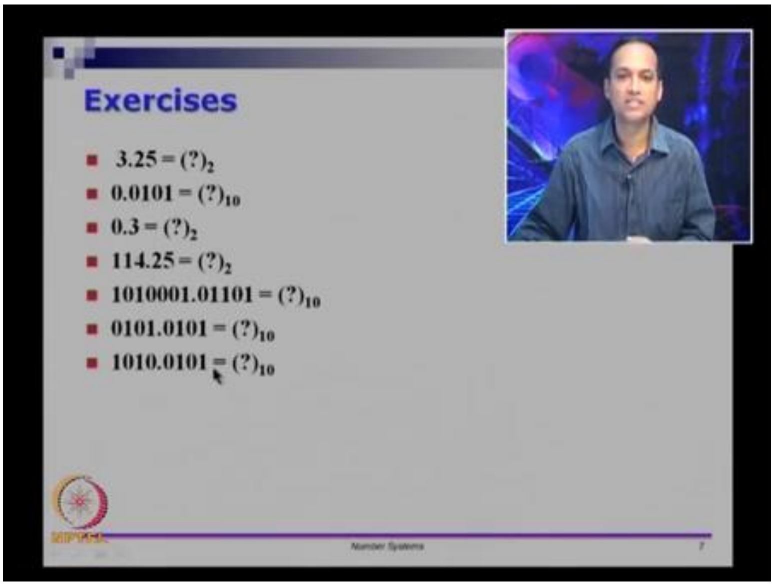


So, if you want a number like this, let us say, it is not just the decimal, it is not just the integer part, I have something, which is a combination of both. If I have both the left and right side, take the left part alone, do as I said before. So, take it repeatedly, divide by 2 until you get quotient equal to 0. Read the remainders from bottom to top. And for the right part, keep multiplying it by 2, keep noting down what is on the left side, but only multiply the fractional part till you hit 1.0 and note it down from top to bottom. And so you can notice, that when you read from bottom to top, it is 10110 you have that here and when you read from top to bottom, it is 1101 you have that. So, you have to be a bit careful about doing this.

So, you, this quotient when you get 0, you should ignore it and the fractional part will always start with a 0, you should ignore it. Because, so when you look at any fractional part, you will always write it as 0 point something, right. The very first 0, you should always ignore it. So, you start from the next one and start noting things down from top to bottom and the concatenation of those two gives me the decimal value or binary equivalent of the decimal value, 22.8125.

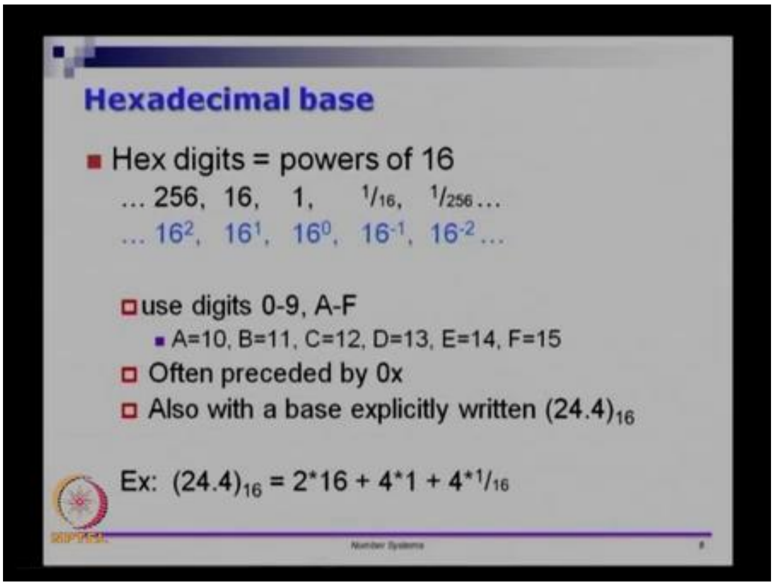
So, again as I said, you go and start putting weights from here. So, this is 2 power minus 4, this is 2 power minus 3, 2 power minus 2, 2 power minus 1, 2 power 0, 2 power 1 and so on. You add all of that, it should eventually add to 22.8125. I suggest that you go and try it out yourself. So, this side is up and that side is down, you just remember that.

(Refer Slide Time: 11:06)



So, these are several exercises that I would like you to look at. So, 3.25 in decimal is what in binary? Then, 0.01 and 01.0101 is by decimal and so on. So, the left side, so the right side says to what you should be convert into the left side is the other one. So, the right side is decimal, left side is binary, left, if the right side is binary, left side is decimal. So, try this out and ensure that you get the answers for it.

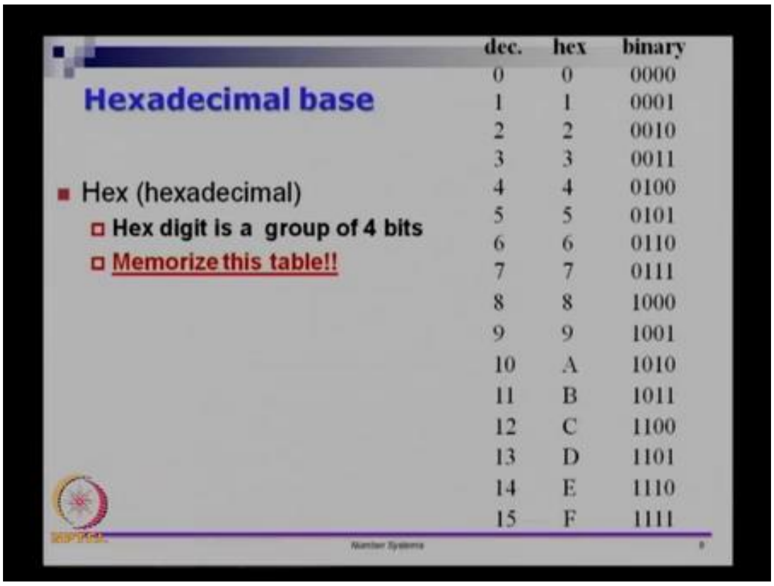
(Refer Slide Time: 11:35)



So, now let us switch to another base system called the hexadecimal base system. The hexadecimal base system is, we instead of using 10s we use 16th. So, 16 is the base. So, clearly, if you want to write hexadecimal, you can use the symbol 0 to 9 to indicate 0 to 9, but number 10 is also a part of the hexadecimal system. We use, for 10 we use A, for 11 we use B and so on because we want to be able to write it down when as humans we want to be able to read it and write it down. So, we want to see how to do that.

So, let us see a quick example first before we get into the details. So, let us take this example, 24.4. 24.4, if it is written in hexadecimal, then it is as good as 16 into, so 4 into 16 power 0 plus 2 into 16 power 1. So, that is for the left side. So, that is, 2 into 16, 32 plus 4 into 1, 4, so that is 36. And right side we have, 4 into 16 power minus 1 or that is 1-4th, so that is 0.25. So, 24.4 in the hexadecimal is same as 36.25 in decimal. We saw this earlier.

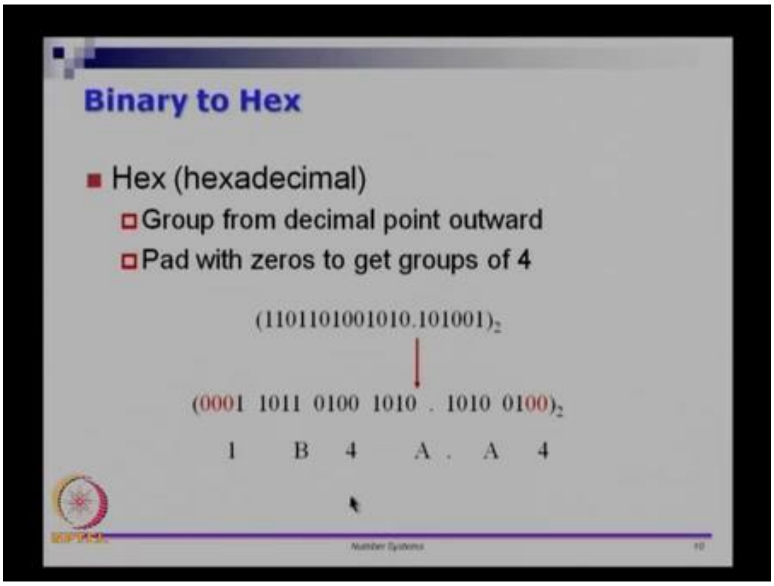
(Refer Slide Time: 12:53)



So, remember and memorize this table. Hexadecimal digit is a group of 4 bits. So, for decimal 0 to 9, you will see that the binary representation is as we expected, so number 10. So, 1010 is decimal 10, but we will represent that with one hexa- digit called A and 1011 will require two digits in decimal, but it actually uses only one hexadecimal digit. So, these are hexadecimal digits, these are all single digit numbers. If you notice, in hexadecimal these are all single digit numbers.

So, if you want to write 10 in hexadecimal we use the symbol A, 11 we use B and so on, up to 15 occupies only one digit. The moment you want to represent number 16, we need two hexadecimal digits. We need two decimal digits also, 1 and 6 and we need two hexadecimal digits. 16 has a representation 1 followed by a 0. You have to remember this table and memorize it is quite useful to memorize this table.

(Refer Slide Time: 13:56)



Then, for binary to hexadecimal, it is very easy to do that. You group from the decimal point outward, but pad with 0s to get groups of 4. So, what it essentially says is, you start from the decimal point and you move outward. Outward meaning, you move left for the integer part and right for the fractional part. So, you start from here and you take four bits at a time and group them. So, this 1010 is a group of four bits, then this 0100 is a group of four bits, then you have 1011 as a group of four bits and you will have just 1, 1 in this example left out. You pad this 0s to the left of that. Similarly, you start from the decimal point, you take four bits at a time and put them here, 1010, you put it here and 01 you put it here and you pad two 0s.

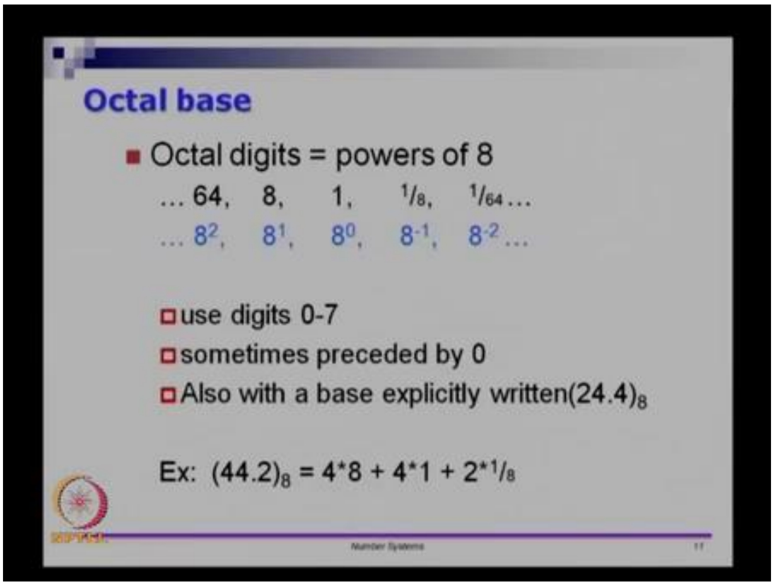
So, this number is the same as this number because you, you know this from decimal side also, that padding 0s to the integer in the front is not harmful. It is not going to change the number. So, if I say, 15 versus 015, in decimal 0 has no weight anyway, it is 0 into 10s square 0. So, it is similarly after a decimal point I can actually introduce as many 0s as I want after my digits are over. So, in this case my digits are over here, I can add as many 0s as I want, it does not matter, however we will pad it, so that it gets to nearest four.

So, in this case we padded three 0s and we stopped. In this case, we padded two 0s and we can stop because we have the hexadecimal representation for this. So, let us see what the hexadecimal representation is. The left side 0001 is 1 itself, 1010 stands for number

11 or it is actually B in hexadecimal representation, 0100 is 4 and 1010 stands for 10 or A. Again, 1010 here stands for 10 or A and 0100 is 4. So, we write that down. So, this is 1, this is B, this is 4, this is A, this is A and that is 4. We can write it as 1B4A.A4 base 16. So, this binary number that you see here is the same as 1B4A.A4 base 16.

So, one thing you can notice is, we really did not do conversion of weights and so on. There is a nice property which we get for base 16. You can start from here and just group them in 4 and you can get this. You do not really need conversion like what we did for decimal to binary and binary to decimal. So, think about why grouping like this is actually valid and it is ok to do so. Go and think about why.

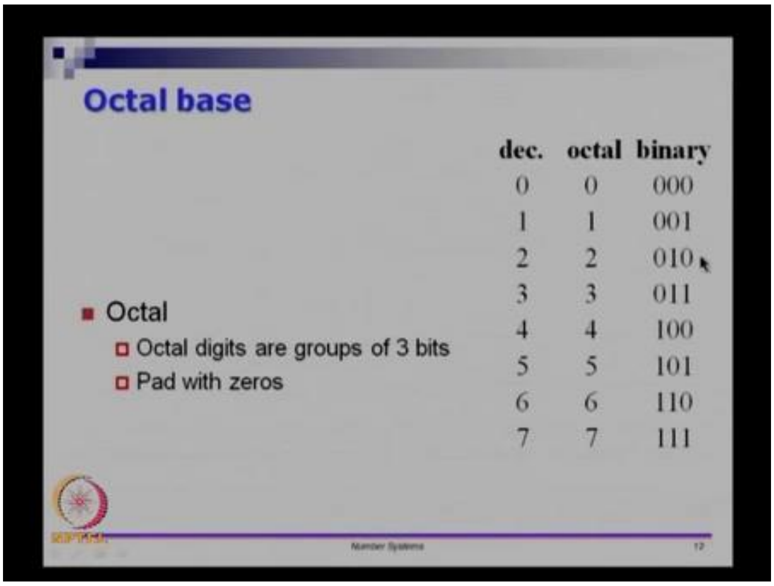
(Refer Slide Time: 16:36)



Then, there is another base system that is also useful in computers that is actually octal. In octal, the base is actually power 8. So, if you are using power 8, then the number 8 itself cannot be there in base 8 system. So, number 8 cannot be there. We can have numbers from 0 or digits from 0 to 7, number 8 cannot be represented or we cannot write number 8 in base 8 system just like number 10, right. You need two digits in base 10 system or decimal system. You do not write them with one symbol, you actually use two symbols, 1 and 0. For 8 you will require two symbols actually, 1 and 0 for octal system. So, we use digits 0 to 7 and sometimes we precede with 0 to indicate the octal number as opposed to decimal number or a hexadecimal number. So, more often than not, I will try and use this explicit specification, 24.4 base 8.

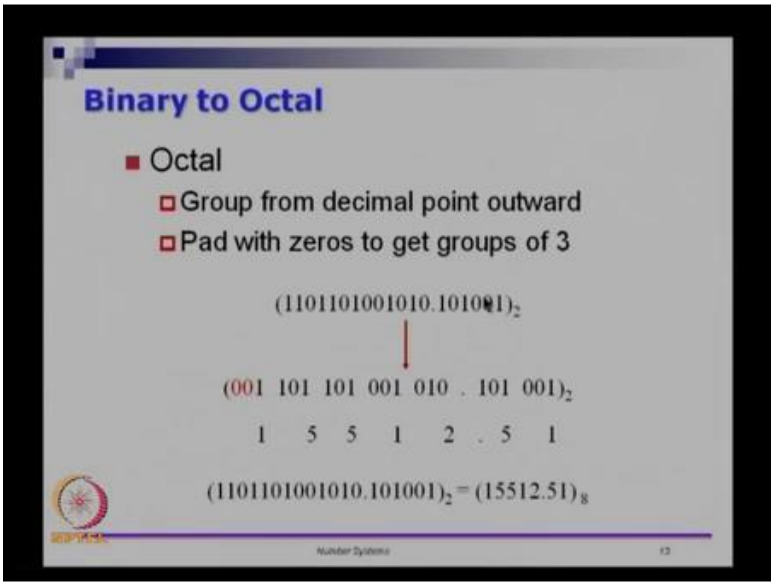
Let us see what 44.2 base 8 is. So, that is 4 into 8 power 0 plus 4 into 8 power 1, that is actually 36 and the right side is 2 into 8 power minus 1, that is, 1-4th. This is actually our same old 36.25.

(Refer Slide Time: 17:49)



And this is a table that you can remember. If you have three 0s, right, you have them and you remember them as 0 to 7. The decimal and the octal digits that we use for 0 to 7 have the same. You do not, you are not exceeding 10, so it is not a problem for octal bits.

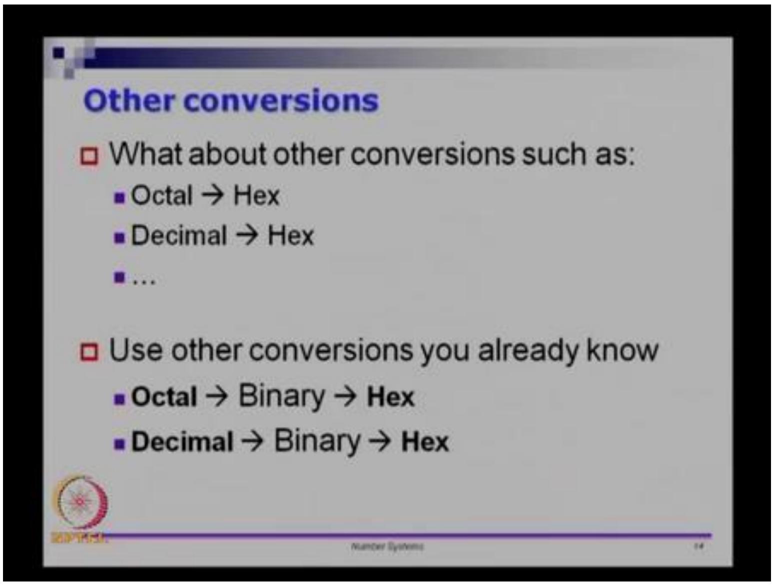
(Refer Slide Time: 18:06)



So, binary to octal is as we did for binary to hexa, only that we group in terms of three. So, you start from here and start grouping in terms of three to the left side and 3 to the right side and once you group it, you may have to do padding. In this case, the leftmost thing needs padding of 2, but we have 6 binary digits on the right side. So, we do not need to pad anymore on the left side. We were two short of a padding them and multiples of 3.

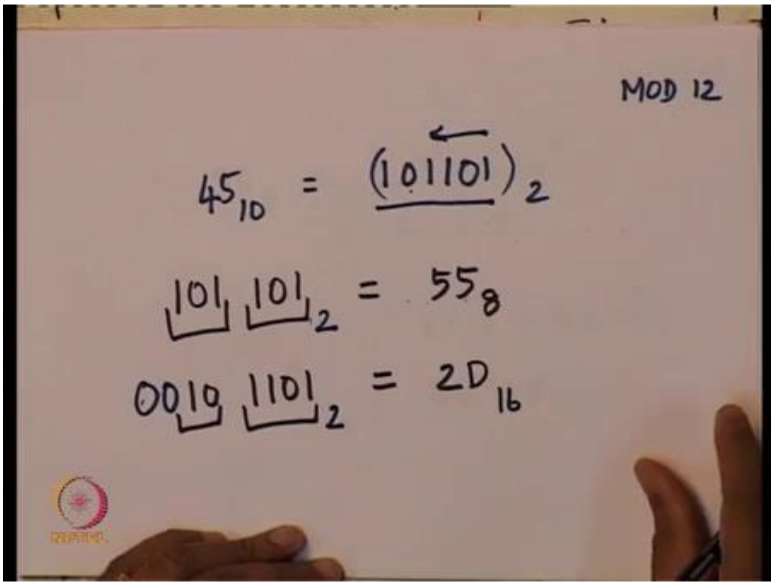
So, there are 13 digits on the left, 13 bits on the left side, 13 is not a multiple of 3, so we need 15 of them and once you have that, go back and put the numbers directly. 001 is 1, 101 is 5, 101 is 5, 001 is 1, 010 is 2, 101 is 5 and 001 is 1, you write that down and you put a base 8 there. So, 15512.51 base 8 is the binary equivalent or is the octal equivalent of this binary number here.

(Refer Slide Time: 19:07)



So, other conversion is straight forward. If you want to do octal to hex or decimal to hex, you can actually try other conventions that you have in place. So, you can convert from octal to binary to hex or decimal to binary to hex and so on. So, I have a small example here on the sheet of paper.

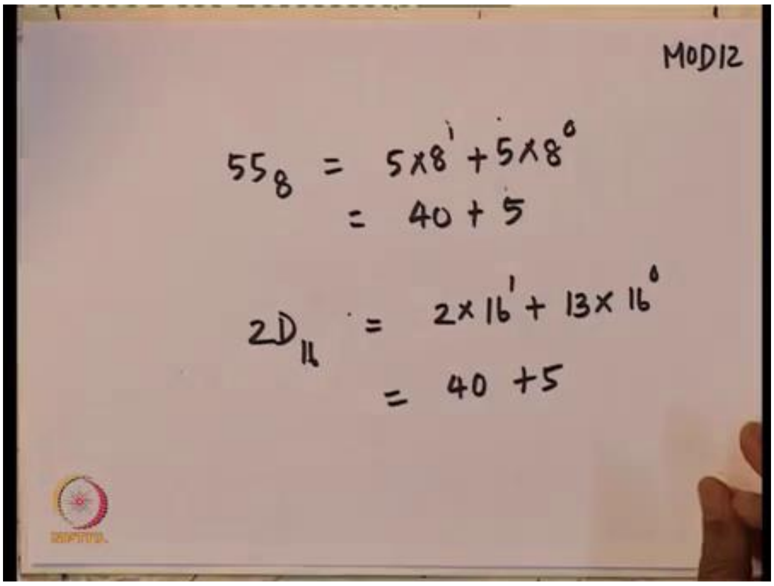
(Refer Slide Time: 19:28)



So, 45 base 10, we already did this earlier, 45 base 10 is 101101 base 2. So, I will take 101101 and group them from. So, there is no decimal point here. So, I can start grouping from this 1, 101 gives me this group and this 101 gives me this group. So, if I read this as, so 101 is 5 and 101 is 5. So, 101101 base 2 is 55 base 8.

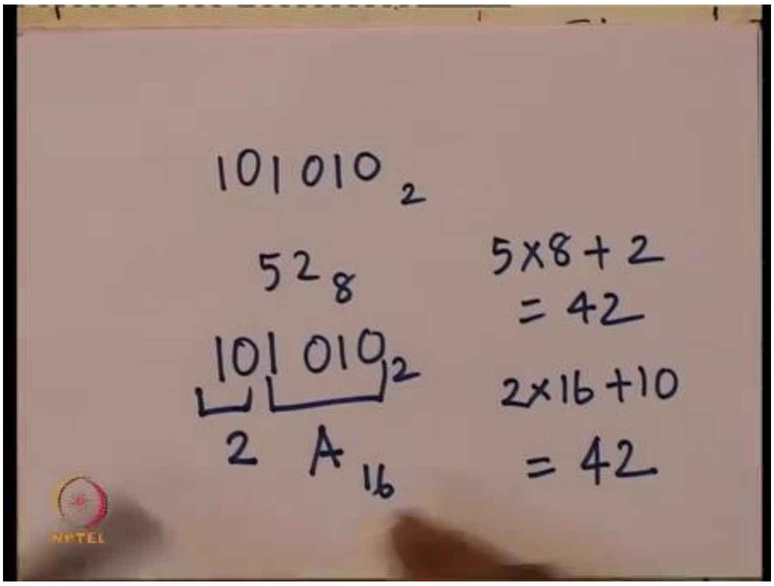
If I want to represent the same thing in hexadecimal, I start from the left, from the rightmost side and move to the left. So, I get a padding of four, it is 1101 and I have two bits here, 1 and 0. I can imagine that that is padded with 00. So, this one is number 2 and this one is symbol D. So, you have 2D base 16. So, 101101 base 2 is 2D base 16.

(Refer Slide Time: 20:32)



If you want to check whether it is correct or not, let us see this we start with 55 base 8. 55 base 8 is 5 into 8 power 1 plus 5 into 8 power 0 that is 40 plus 1, which is 45. Similarly, 2D base 16 is 2 into 16 power 1 plus D stands for number 13, 13 into 16 power 0. So, that is 32 plus 13, which is 45? So, this way of grouping from start at the decimal point, move to the left and group it three at a time for octa, and four at a time for hexa-, is actually a valid way to do things. In fact, if you want to go from octal to hexa-, you, you go and write the binary representation for octa, then group in terms of four, you will get hexa-.

(Refer Slide Time: 21:22)



So, for example, let us say I want to convert this. So, this is in base 2. Let us say, instead I had only the octal representation 52 base 8. Let us say I only had that, forget this for a while, let us say, I had only 52 base 8, then I can go and write the binary representation. So, 2 base 8 is, we can write it as 010; for 5 we can write it as 101. And now, that is in binary, we can actually group in terms of four and write it as hexa.

So, 2A base 16 is same as 5 to base 8, we can verify that. So, 5 to base 8 is 5 into 8 plus 2, that is the same as 42 and 2A base 16 is 2 into 16 plus 10, that is the same as 42. So, it is a fairly simple way to convert from octal to hexa- or from hexa- back to octal. We use the binary root because conversion is easy to binary and then in binary do a packing of three or four appropriately.

So, this brings me to the end of this week lectures and we have seen quite a bit of things in this week. We have, we started with K-maps on two variables, three variables, four variables and so on and we also looked at how to use K-maps as a visual infrastructure to help us in doing Boolean minimization. So, probably by now you understand that this graphical representation is much more powerful because it is easier for human beings to do than this Boolean algebraic minimization. We also saw a few other things, namely multiple output minimization and number systems and so on. The quiz, that you will have this week will cover all of this material.

So, this brings me to end of week two and thank you very much, I will see you next week.