Computer Organization and Architecture: A Pedagogical Aspect

Prof. Jatindra Kr. Deka
Dr. Santosh Biswas
Dr. Arnab Sarkar
Department of Computer Science & Engineering
Indian Institute of Technology, Guwahati

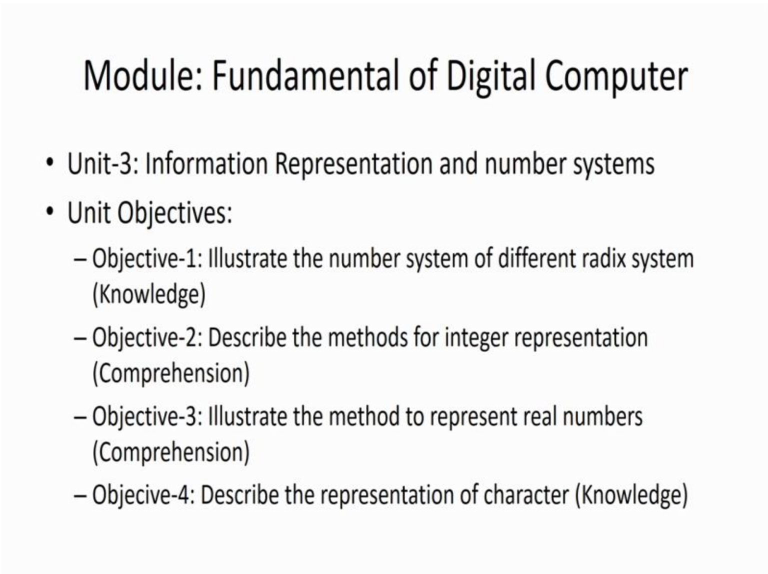
Fundamentals of Digital Computer

Lecture – 03
Information Representation and Number systems

Hello everybody, welcome back to the online course on Computer Organization and Architecture. Now we are in the module of fundamentals of digital computer now we are in unit 3 and unit 3 is basically related to information representation and number system. In this unit we are going to see how we are going to represent information for digital computer and the number system that we are going to use for our arithmetic.

So, first of all we are going to state the objective, what is the unit objective of this particular module?

(Refer Slide Time: 00:59)

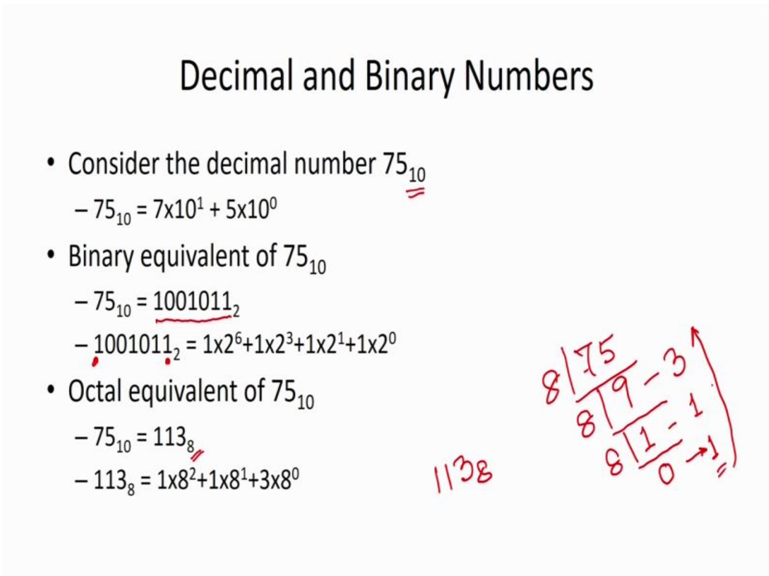


So, the first objective is illustrate the number system of different radix system. So, this is in the knowledge level. So, we are going to discuss everything in the knowledge level. So, that we will have the knowledge about the number system and we can use these things in digital computer.

Objective 2, describe the method for integer representation. So, this is in the comprehension level. So, when we see these things you will be able to understand how we are going to represent integer in computer system. Objective 3, illustrate the method to represent the real numbers. This is also in comprehension level we are going to see how we are going to deal with real numbers in computer. And objective 4, describe the representation of character this is also in knowledge level we are going to see how we are going to represent character in computer because we have to deal with data processing and many a time we are going to work with string manipulation.

Now, first we are going to talk about the number system.

(Refer Slide Time: 02:01)



I think all of you know about the decimal number system where the base of this number system is 10 or in general we say radix system. Consider an example of decimal number say 75 and I just ask you if you have written as 10, it means that the base of this number system is 10. So, now, 75 how we are going to evaluate the value of 75, it is $7 \times 10^1 + 5 \times 10^0$, so 70 + 5 this is 75.

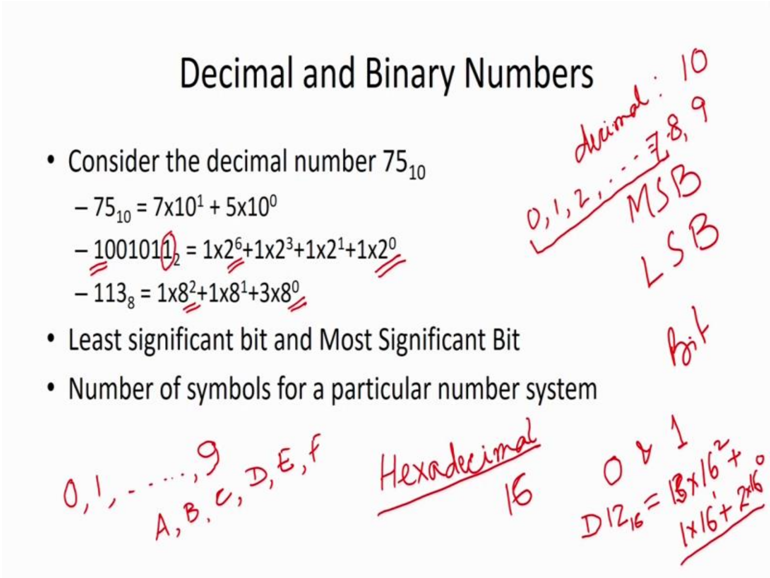
Now, what is the binary equivalent of this particular 75? Now whatever we are going to do eventually we should get 75. So, I am not going to discuss about conversion just I am giving an example. So, 75 in decimal number system will be represented with 1001011 in binary number system. So, when we talk about the binary number system then the radix or base of the number system is 2 which is given in the subscript over here.

Now, in base 10 system it is of radix 10 and it is a multiple of 10, but in case of your binary number system it is multiple of 2. So, in that particular case now we are saying that fifth is the 0th position, 7 is the unit position of our decimal number system. So, similarly here also we are going to have this position this 1 is the 0th position like that 1 2 3 4 5 6. So, this 1 is in the 6th position. So, ultimately what is the equivalent value of this binary number? $1 \times 2^6 + 0 \times 2^5$ which is $0 + 0 \times 2^4$ which is $0 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$. So, this is $2^6 = 64$, 2^3 is 8, 64 + 8 72, 2^1 is 2, 72 + 2 is 74 and 2^0 1, so + 1, 75. So, eventually we are getting the decimal equivalent 75. So, in this way we can represent every numbers every decimal numbers in binary representation.

Like that we are having different radix system and as for example, I am giving another base system which is your 8 base is 8. So, now if you consider this 75 in decimal number system that equivalent octal number system, so base 8 number system is known as your octal number system it is 113 so that means, $1 \times 8^2 + 1 \times 8^1 + 3 \times 8^0$. So, 8 square is 64 + 8 = 72 + 3 =75. So, this is the way we can have a conversion from one number system to another number system.

Decimal to any other number system is very easy very simple what we can do, simply divide that number by that particular base. So, in that particular case 75, so this is in radix 10 or base 10 we want to convert it to octal number system. Then what we are going to do? We will divide it by 8, 8 9s are 64 and remainder is 3 again we are going to divide 9 by 8, 8 1s are 8 and remainder is 1 again we are going to divide it by 8 then result is 0 remainder is 1. So, in that particular case we are going to get 113 in octal number system. So, 75 in decimal number system is equivalent to 113 in octal number system. This is the way we can represent or we can convert every decimal number to any other radix system. So, similarly when we are going for a binary number system then 75 will be divided by 2 and in that particular case now where we are dividing again this is coming in the unit position and like that it will increase. So, it is 113.

(Refer Slide Time: 06:22)



So, now when we are going to talk about a number system we are having two terms called least significant bit and most significant bit MSB and LSB this is basically when we are going to talk about the bit it basically talks about the binary number system we are going to take single bit. But for any radix system we can say it is a most significant digit and least significant digit. As for example, in your 113 in base 8, 3 is the least significant digit and 1 is the most significant digit in this particular case because you just see that this 1 is multiplied by 8^2. So, weightage is more and these 3 is multiplied by 8^0 which is one. So, weightage is less, so in that case we having that least significant digit as 3 and most significant digit as 1 for 113 in octal number system. Similarly when we are going to talk about the binary number system we say it is least significant bit and most significant bit.

So, in this particular case the one or bit that we are having a most right hand side is known as your least significant bit because it is weightage is less 2^0 and the one that we are having in the left hand side most left hand side is your most significant bit because the weightage is more over here 2^6. So, this is the convention about the least significant bit and most significant bit.

Now, how many numbers or how many symbols we require for a particular number system. So, it is basically related to the base of the number system. So, if we are using a decimal number system decimal number system in that case base is 10. So, to represent these things we need 10 different symbol and we know that those symbol we are using 0 1 2 3 like up to 9 we are using. So, these are the 10 different symbol that we are using in decimal number system. So, in case of binary number system since base is 2 we need only 2 symbols and that 2 symbols that we are using in binary number system is your 0 and 1. So, the binary number is represented with the help of 0s and 1 only.

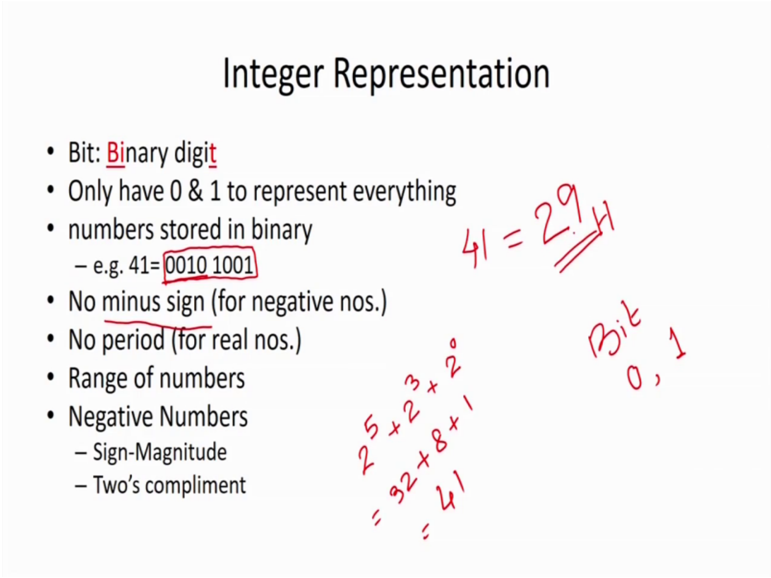
When we come to octal number system in that particular case the base is 8. So, we need 8 different symbol. So, what are the symbols that we are using? We are using from 0 to 7 these are the 8 different symbol we are using in octal number system. Like that we can go for any radix or any base. One important number system that we are having is your hexadecimal. So, in computer system sometimes we represent our information in hexadecimal form. So, in hexadecimal the base is 16, so we need 16 different symbols. So, in hexadecimal system the base is 16 and we need 16 different symbol to represent it. Now, what are those 16 different symbols that we use in our hexadecimal number system?

So, from 0 to 9 we are using those symbol to represent 0 to 9, 0 1 2 like that since it is having base 16 so, we can represent 16 different numbers in this particular number system. So, 0 to 9 10 is gone now you are having 6 number 10, 11, 12, 13, 14 and 15. So, to represent those number in hexadecimal system we use the letter from our alphabet and these letters are your A, B, C, D, E and F. So, this is the symbols that we use in our hexadecimal number system, so 0 to 9 ten digit and A to F six characters, total 16 character.

So, if we write some numbers like that say D12. Say this is a hexadecimal number and we are using the symbol. What is the decimal equivalent of these things? This is your D is your 10, 11, 12, 13. So, this is $13 \times 16^2 + 1 \times 16^1 + 2 \times 16^0$. So, whatever we are going to get this is the decimal equivalent of this particular hexadecimal number. So, conversion from any number system to decimal number system is very easy. So, like that from decimal also we can convert to any number system just by divide by the base of that number. So, this is the way we can just represent our numbers.

Now, we are going to see how we are going to represent integers. So, this is 0s and 1s only. So, we will say whatever we are representing we say it is a bit and bit is nothing but binary digit; that means, we are representing 0 or 1. So, these are the binary digit.

(Refer Slide Time: 11:33)

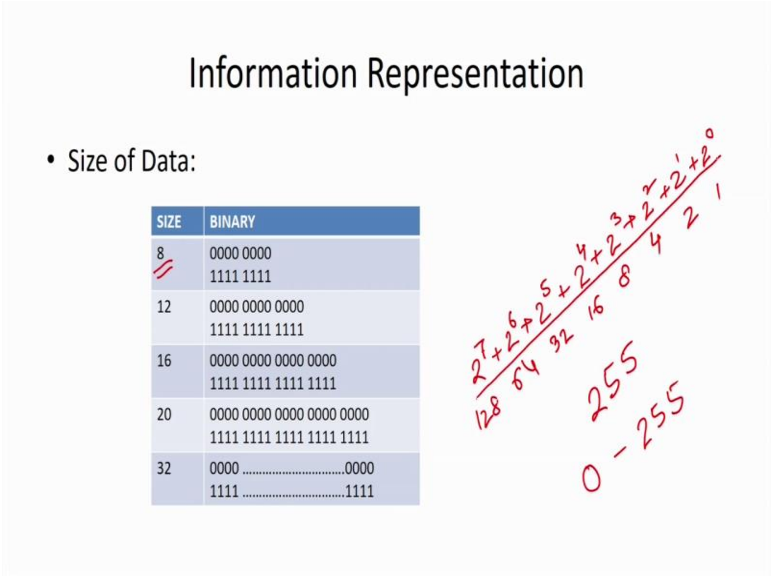


And already we have seen how we are going to write numbers in binary number system. So, for example, here we are giving 41. So, in that particular case, 41 is having binary equivalent 00101001 so that means, we can say that this is these are the 3 ones are coming. So, these are going to have going to give us the effect so, it is basically $2^5 + 2^3 + 2^0$, 0 1 2 3 4 5. So, 2^5 is 32, 2 cube is your 8 + 2^0 is one. So, this is equal to 41. So, this is the way we are going to represent our integer in binary number system.

So, there are some issues now we have to see how to give the - sign to have the negative numbers, how to put a decimal point for real numbers and secondly, we have to see what is the range of numbers. Ok so, basically say if I ask you to do something with pen and paper then you don't have any limitation, but when we are going to work to our digital system or we say at we are going to work with digital computer then always we are restricted by a range and this range depends on the bit number of bit that we are using to represent the number.

So, in this particular case of 41, in this particular case we are using 8 bits. So, when we are using 8 bits; that means, we are restricted by a number up to a particular limit only we can go. So, if we increase the number of bit then we can go for more ranges and. Secondly, we are going to see how we are going to represent the negative numbers there have been two way of doing it one is your sign magnitude and second one is your 2's complement. Just as an example I am giving here is a size of my data.

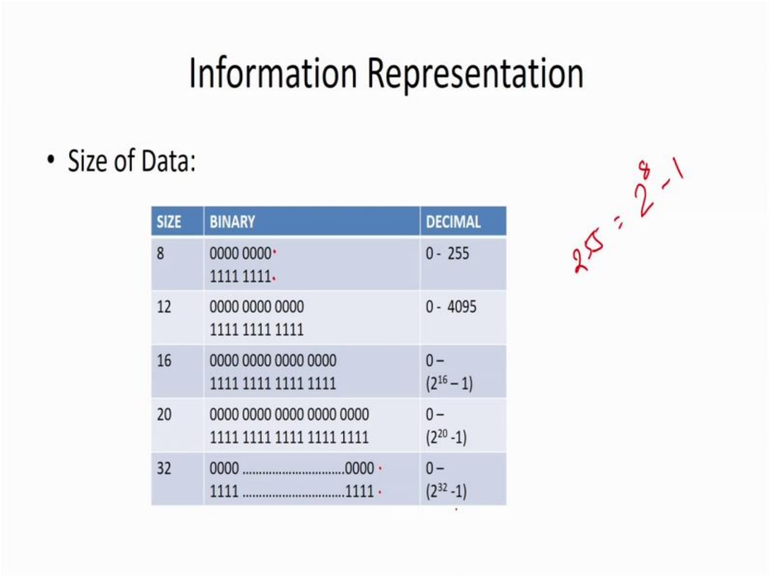
(Refer Slide Time: 13:44)



So, if you are walking with your size 8; that means, you are working with the data size of 8 bit. So, in that particular case the combination may be all 0s to all 1s. So, in that particular case all ones is going to represent can you calculate it this is your 0 to 7; that means, $2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0$. So, this is your 1, 2, 4, 8, 16, 32, 64, 128. We have to be you have to be accustomed with this particular 2 to the power of something because many a time we are going to look it. So, this is I am going to get 128 64 32 16 8 4 2 1 and if I add up all those things I am going to get 255. So, when we are going to work with 8 bit numbers then my range will go from 0 to 255.

Ok, similarly if I am using a 12 bit numbers now my range is going to increase like that if I am going for 32 bit numbers; that means, we are going to work with 32 bits at a time. So, nowadays we used to say that my computer my processor is a 32 bit processor; that means, I can work with a 32 bit numbers. I say that my computer is or my processor is a 64 bit processor then we can handle 64 bit at a time.

(Refer Slide Time: 15:27)



So, in that particular case we can say that these are binary representation for 8 bit this is the minimum number this is the maximum number if we are going to work with a positive line only and if this is 32 bit this is the minimum one this is the maximum one. So, my range is in decimal is for 8 bit numbers it is from 0 to 255 for 12 bit number it will 0 to 4095. So, this is 4095 and when I am going for 16 bit number it is 0 to 2^{16} - 1 basically 255 is nothing but I can say 2^8 - 1, 2^8 is a 256 - 1, 255 and if it is your 20 bit then into go up to 2^{20} - 1 and in case of 32 bit it will go to 2^{32} - 1.

(Refer Slide Time: 16:19)