Microprocessors and Microcontrollers
Prof. Santanu Chattopadhyay
Department of E & EC Engineering
Indian Institute of Technology, Kharagpur

Lecture - 01
Introduction

Microprocessors and Microcontrollers: so they are part of computing systems. Now, computing systems it ranges over a wide range a wide set of applications. Starting from a very small like calculators, then cell phones and very small applications till we have got very high end computing platforms like supercomputers.

Now, while we are talking about say small applications due to their smaller size and space power requirements we have to optimize, and there we will see that microcontrollers are most commonly used. On the other hand when we are looking for computational power, that is, we want systems that will computationally, it will be able to solve a large number of users different types of programs and computations. There microprocessors are used, because they have the flexibility, they have the capability to work at a higher rate generally than the microcontrollers.

Of course, there are microcontrollers which are better, which support high speed applications, but mostly we have got microprocessors for doing this computation job. Now, whatever it is or both microprocessor and microcontroller they are some computing element. In the sense, that they perform computation over some data element that is taken as a input and that data element is represented inside the system in some fashion and then processing is done over that and ultimately it is output to the to the environment.

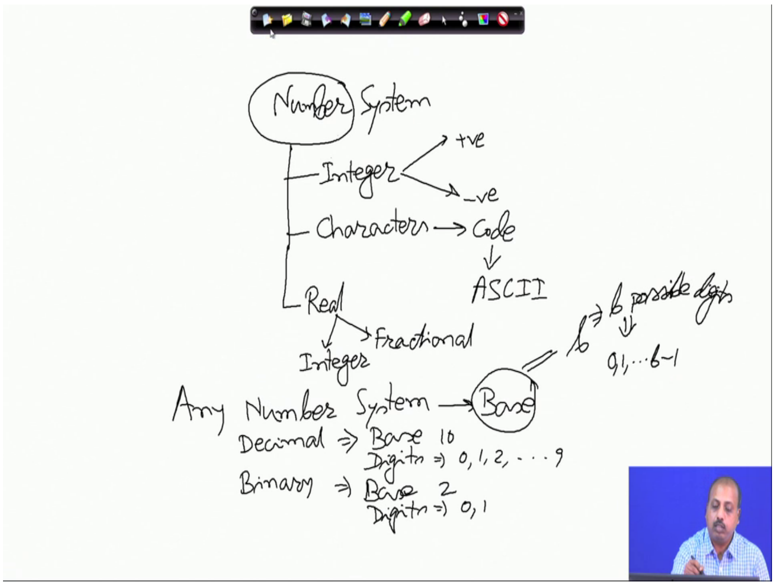
Though we know that the environment is mostly analog like when we are taking signals from a different from the environment the signals are analog in nature, but for the computational purpose it is better if we have digital inputs. These digital inputs, for the digital inputs will help us in the sense that it will reduce the amount of noise that gets introduced in the calculation process and computation can be done in a much better fashion. So, this conversion, that is, this analog to digital conversion, there are some specific modules for doing that.

So, while dealing with the course on microprocessors and microcontrollers we will take the data input as digital only. Of course, we will look into some interfacing later that will convert the analog signals into digital signal, but that is a different issue. For information that is stored and processed by this processor, they are essentially digital in nature.

So, for storing the any value in the form of a digital quantity we need to store we need to have some numbers for storing them. So, for example, if we are storing say the voltage value of a point, so, it is few volts or say millivolts or something like that. That value has to be stored, some number has to be stored in that sense or if we are storing some name string also representing the name of a person, so, that is also stored inside the computer in the form of some numbers.

So, this representation is a very important issue and that representation, how are you going to represent a number in the computer system is a fundamental part, like how these processors they will do the processing on top of that. So, to understand that part, at the beginning we will just recapitulate the number system representation for various types of a numbers that we have. So, to start with we will look into the number systems.

(Refer Slide Time: 04:01)



So, number system, this talks about how a number is stored inside the computer system. Now, as you know when I say number, so, this number can be of different type; like it can be integer number and this integer can again be positive integer or it can be negative integer and otherwise, similarly we have got characters. They are used for storing character strings, like say the name of a person then some other address and all that. So, that way we have got the characters stored. So, characters are also stored by means of some code and this code is there is a very popular code that you may be familiar with which is known as ASCII code where each character is given some integer code and ultimately inside the computer what is stored is nothing, but this integer code. So, it is not processed as a character inside the computer, it is stored as number only.

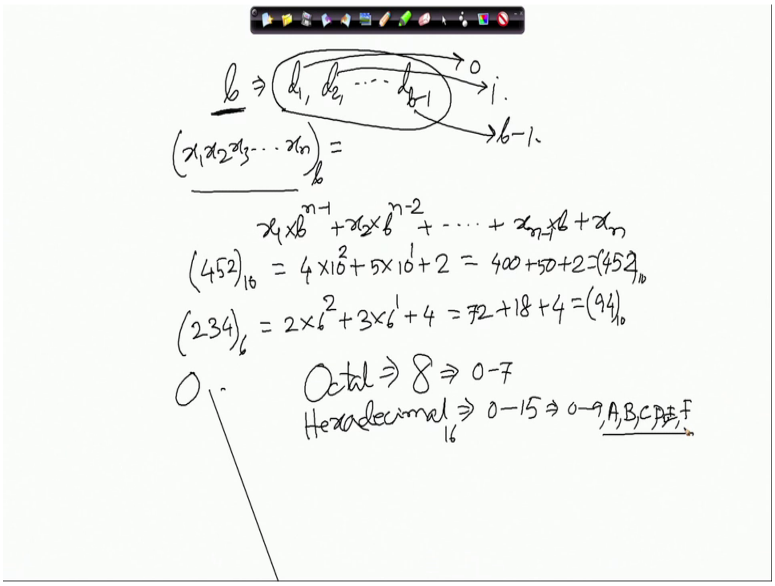
Then, other class of numbers that we have is the real numbers, that is, real numbers will have 2 parts again; one is the integer part and the other part is the fractional part. So, this is known from our school days, from our numbers that we are handling within our mathematics classes. So, these are the things that we need to represent inside a computer.

Now, how are you going to do that? So, that is another issue and as we know that in a computer system, numbers are stored in a binary format. So, we will try to understand what is this number system, in general, can we have a generalized, can we have a very generalized number system of arbitrary base and values etcetera.

So, if we talk about any numbered system, it has got one important part in it, which is called the base of the number system. So, base means what is the highest value that a single digit of it can be present. So, it is plus or how many digits, how many types of digits can be there. For example, we are familiar with the decimal number system; in the decimal number system we have got this base value as 10 and the symbols or digits we have 0, 1, 2 up to 9. So, we have got 10 different digits and it is represented the base of the number system is 10 or if you have consider the binary number system as we know there the base value is 2 and the digits are 0 and 1.

So, in general, if we have this base value is equal to b then; that means, there will be b possible digits in the number system. So, we have got b possible digits. These digits are marked as 0, 1 going up to b minus 1.

(Refer Slide Time: 08:03)



So, if we go for any number, say with the base b and it has got the digits like say d 1, d 2, so, these are the possible digits up to d, b minus 1. So, these are the possible digits, then we can have a number in this number system which is say x 1, x 2, x 3, x n to the base b, so, it is represented like this to the base b. So, this is the first digit is say, the value of the first digit is d 1, value of the second first symbol is d 1, second symbol is d 2, third symbol is d 3 like that. So, and these values are given by the powers of b like. So, actually this d 1 to d b minus 1, they are not arbitrary values. This d 1 is always 0 then d 2 is 1. So, in this way this d minus 1 is the value b minus 1. That way it goes.

So, we have got a, if this is the number x 1, x 2, x n to the base x 1, x 2, x n up to b. So, this is given by x 1 into b to the power n minus 1 plus x 2 into b to the power n minus 2 plus it goes this way plus x n minus 1 into b plus x n. So, if we take an example suppose we have got a number 452 to the base 10. So, how does it corroborate with this definition. It is 4 into base is 10 to the power of here value of n is equal to 3, so 10 to the power of 2 plus 5 into 10 to the power of 1 plus 2. That way it is 400 plus 50 plus 2. So, that is 452.

So, in this way whatever be the base, accordingly we can think we can find out what is the corresponding number in the decimal system. For example, if I take any arbitrary number say 234 to the base 6, what is this number? So, this number is 2 into 6 to the power 2 plus 3 into 6 to the power 1 plus 4. So, that is 72 plus 18 plus 4. So, that is 72 plus 18, 90 plus 4, 94 in the decimal number system. So, this is the decimal number system.

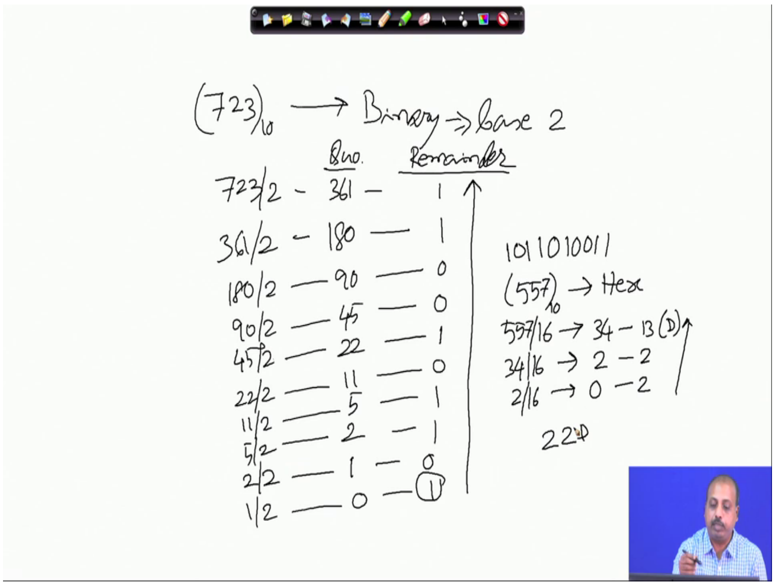
So, this way numbers given in any number system, we can find out the corresponding value in the decimal number system and of course, we can represent any number in any number system. What I mean is, in any value that is given to me, in decimal number system I can represent it in some any other desirable number system. So, that can be done, but something to be noted is that, if the base of the number system is b then the digits that we have must be 0, 1 like this up to b minus 1. So, it cannot be that it has got a digit whose value is more than b. So, that is not possible then it is not a valid number system.

So, now how to convert, how to get from 1 number system to another number system? So, before that, you are also familiar with say octal number system, like the octal number system where the base of the number system is 8 and you know that the digits that we have there are 0 to 7. Then we have got hexadecimal number system where this hexadecimal number system the numbers can be the base value is 16 and the digits can be 0 to 15. Now, it is difficult to represent 15, so, what is done is we introduce some new symbols. So, 0 to 9 is there and after that we say A standing for 10, B standing for 11, C standing for 12 like that D, E and F.

So, these are the numbers that we these are the digits that we have in the hexadecimal number system, this way you can think. So, this A, B, C, D, E does not matter. So, we have to remember; what is their corresponding decimal value and you can introduce any arbitrary symbol for them. So, that it is not mandatory that you have to write in terms of A, B, C, D, E, F like that you have to remember the corresponding decimal values.

So, let us next consider how to convert one number, a number from one number system to another number system. So, the trick is when you are converting from say decimal number system to a number system of different base, we go on repetitively dividing the number by that base. So, we will take an example.

(Refer Slide Time: 13:56)



Suppose, we have the number say 723 in the decimal system and I want to convert it into say binary number system. So, binary number system the base of the number system is 2, what I do is I just go on repetitively dividing 723 by 2 and note down the reminder values.

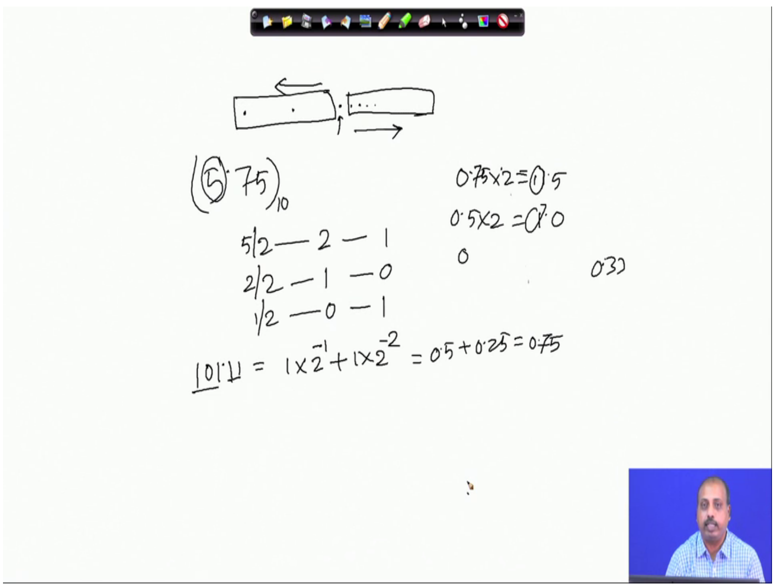
So, 723 divided by 2. So, if I note down the quotient part and the remainder part, this is 361 and this is 1, then this 361 if we will divide by 2 again and we will see like what is the value coming. So, 361 divided by 2, this will give me reminder as quotient as 180 and reminder again as 1, then this 180 divided by 2, it gives me 90 as the quotient and 0 as the reminder. 90 divided by 2 gives me 45 as the quotient and 0 as reminder. 45 divided by 2 gives me 22 as the quotient, 1 as reminder. 22 by 2 is 11 as quotient, 0 as remainder and the 11 by 2 gives me 5 as quotient, 1 as reminder. 5 divided by 2, this gives 2 as quotient 1 as reminder. 2 divided by 2 it gives 1 as quotient and 0 as remainder and then 1 divided by 2, this gives 0 as quotient and 1 as remainder.

Now, while representing the number, you see that this last digit, this particular digit it has come after so many divisions by 2. This must be the digit of most significant, most importance. It is the most significant digit. While writing the number in binary number system we will write in this sequence; so first 1, then 0, then 2 once, then again 0, then 1, then 2 zeroes, then 1. So, this is the representation of 723 in the binary number system. So, the same way, you can represent any number in any other number system.

For example, suppose we are interested to represent the number 557 in hexadecimal number system. So, this is base 10 we want to convert it into hexadecimal number system, that is, base 16 number system. So, what we have to do is, the same thing 557 divided by 16. So, this will give you quotient as 34 and reminder value as 13. 13 in hexadecimal number system is represented by the symbol D. So, this is D, then this 34 divided by 16, so, this will give you 2 as the quotient and 2 as the remainder and then this 2 when you divide by 16, it gives 0 as the quotient and 2 as the reminder. So, the number that you have is again in this sequence 22D is the number in the hexadecimal number system. So, this is the representation of 557 to the base 10 to hexadecimal number system.

So, this way you can convert any number from decimal number system to other number systems by dividing it by the base of the number system.

(Refer Slide Time: 17:57)



We can also convert in some other the fractional numbers is that the real numbers you can convert real numbers also. But in real number what happens is, that if this is the integer part of the number, after that we have got a decimal and after that you have got the fractional part.

Now, in this case in the integer part as you are going from this side to this side the weight of the numbers are in weight of the digits are increasing. A digit here is more important than a digit at this point and that is going by the power of this. On the other hand on the fractional part, the numbers with the digits which are close to the fraction point that is this point decimal point, they are of more importance. So, as you are going away from this decimal point the significance of that digit decreases.

So, you can say if we have a number say 5.75 in decimal number system and we want to convert it into binary number system. Then first of all, this 5 has to be converted into binary number system. So, 5 divided by 2, we have got 2 sorry you have got 2 as the quotient and 1 as the remainder; then 2 divided by 2, you have 1 as the quotient and 0 as the reminder and then this 1 divided by 2, you have 0 as the quotient and 1 as the remainder.

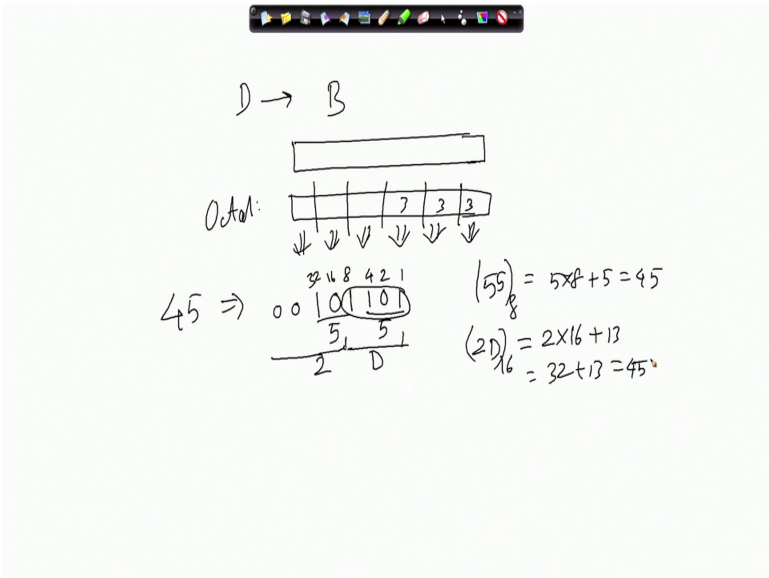
So, for the 5 part I have got the digits 101, then comes the decimal point and now, this 0.75. This 0.75 will instead of dividing by the base, we will multiply by the base. So, 0.75 into 2 that will make it 1.5 and in that 1.5 we will take 1 as the significant number. So, this multiplied by 2, this gives me 1.5. So, out of that this 1 is important, so, this 1 is taken and this 0.5 is again multiplied by 2. So, you get 1.0, again that 1 is taken and then it becomes 0.

So, after that this value has become 0, there is no point here doing the multiplication again. So, here the fractional part is represented as 1 1 and you see that this really represents the number for 5.75 because the left side before the decimal point represents 5 the right side, the representation is 1 into 2 power minus 1, because that is the base of the number system is 2 and. So, we go by 2 power minus 1 plus 1 into 2 power minus 2. 2 power minus 1 that is half that is 0.5 plus 0.25. So, this is 0.75.

So, this way you see that this number if we multiply the fractional part by the base of the number system take the integer part of it as the next significant digit and go on multiplying the remaining fractional part by the base of the number system. So, this way we can get the entire fractional part represented in the form of that particular number system. So, basically, this if the number is such that it cannot be represented precisely in that number system for example, if the fractional part is 0.33, so, if you go on multiplying 0.33 by 2, it will never come to 0. So, it will go on having some reminder fine. So, that is there, but it will be.

So, after some time what happens is that there when you are storing that number in a computer system you have got a finite word size. So, you cannot represent the number beyond some accuracy. You have to stop at that point and whatever it is, you have to take that representation. Though it will definitely never be equal to 0.33 or whatever be the number of digits that you have taken in the consideration. So, this way we can convert from one number system to another number system.

(Refer Slide Time: 22:29)



.

There are other shortcut methods by which you can convert a number to that the hexadecimal or octal number system because if you have a number a decimal number D and if you have the corresponding binary number B; suppose, this is the binary representation of the number, so, what you can do for octal? For octal number system you can divide it in terms of bits of groups of 3 bits. So, each of them is a group of 3 bit, 3 bit group.

So, whatever it is, now, you can convert each of this 3 bit number group into the corresponding octal digit and that will be the octal representation. For example, if I consider the number say 45 in the binary number system, this is 1 0 that is this is 1 0 then 1 this is the 2 power 32, 16, 8, 4, 2 and if we take that way. So, the forty, this is also 1 this is 0 this is 1. So, the 1 0 1 1 0 1 is the binary representation of 45.

Now, as I have said to convert it into octal number system, you have to group it into 3 bits. So, these 3 bits there that gives me the number of digit 5, this gives me the 55. So, 55 to the base 8, you can verify it. It is 5 into 8 plus 5 that is equal to 45. So, this 55 in octal system is 45 in the decimal system.

The same number if you want to represent in hexadecimal form then you have to take 4 group 4 bit group, this is one 4 bit group. Now, after that, this has got only 2 digits in it, 0 and 1. So, you have to augment it by 2 more zeroes. So, this is the next digit that we have. So, this part represent 8 plus 4, 12 plus 1, 13 is D and this part represents 2. So, that is 2, so, this is 2 D.

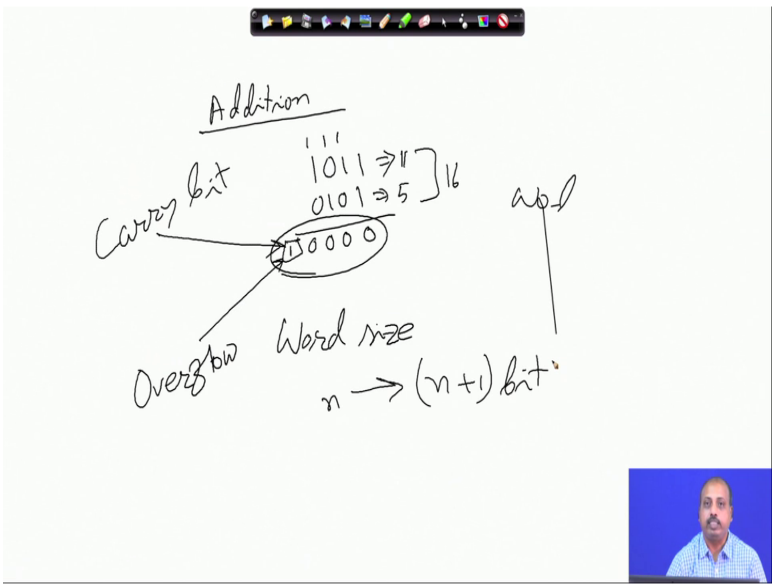
This is 2 D in the hexadecimal number system. So, you can convert this 2 D and check that it is really representing the hexadecimal number system, 2 into 16 plus D; D is 13, so, that is 32 plus 13 equal to 45.

So, this way by using this number system you can any number system you can use they are all equivalent, you can represent numbers in a proper format. Of course, you can there is another representation of these real numbers which we do not consider here that is the floating point representation. So far, whatever representation of a real numbers we have shown the decimal point was fixed. So, they are known as fixed point number system. There is a floating point representation as well which is a bit complex and most of the microprocessors and microcontrollers that we will discuss; they will not have these floating point capabilities.

Normally, it is they are provided by having some coprocessor along with the microprocessor or microcontroller. So, we will not go into that part. So, will be talking mostly about integer numbers and some cases we may talk about these real numbers and that also in the fixed point notation.

So, the next important thing that we will look into is how to do this operation, like how to do this addition, subtraction, multiplication type of operation in basic number in basic number systems.

(Refer Slide Time: 26:14)



So, the first most important operation that we have is the addition of 2 numbers. Now, addition of 2 numbers from our school day knowledge in mathematics, we can do that. Basically, if I have got 2 numbers say 1 0 1 1 and say 0 1 0 1. So, you can add them bit by bit, I am talking about binary number system in the here. So, this sum on 1 plus 1 is 0, with a carry generated as 1. So, 1 plus 1 is 0 with a carry generated as 1, 1 plus 1 is 0 they carry generated as 1 and this is also 0 and there is a carry generated which is 1. So, let us check. So, this is 8 plus 2 plus 1, that is 11 and this number is 1 0 1 that is 5. So, 11 plus 5 is 16.

So, you see that this is the representation of 16, but of course, there is a concern like when I am taking this the number like 11 and 5, I have taken 4 bit representation, but the result that I have got is a 5 bit representation. So, it cannot be the number 16 cannot be represented in a 4 bit number system.

So, in computer system, this is a very important problem because in computers we cannot give arbitrary length to individual numbers. The addition cannot be done over arbitrary sized numbers, so that, there something we call the word size of the processor, sorry it is called the word size of the processor and this word size tells that; what is the data size on which this the processor will operate. So, if the processor operates on 4 bit data then this becomes a problem, so we cannot have a 5 bit result stored there. So, we cannot have this any results coming out which is 5 B.

So, this creates difficulty and this situation is known as the problem of overflow. So, the result is more than the 4 bit representation and in many a time we try to accommodate this extra one bit that is generated by means of some additional storage which in a processor we normally call a carry bit. So, there is one extra space devoted for storing this extra bit that is generated and this is often known as the carry bit. But, you must understand 1 thing that if we are adding 2 n bit numbers then the result can be of n plus 1 bit not more than that. If you are adding 2 n bit number the result can be of n plus 1 bit not more than that. That justifies keeping a single carry bit for doing the addition.