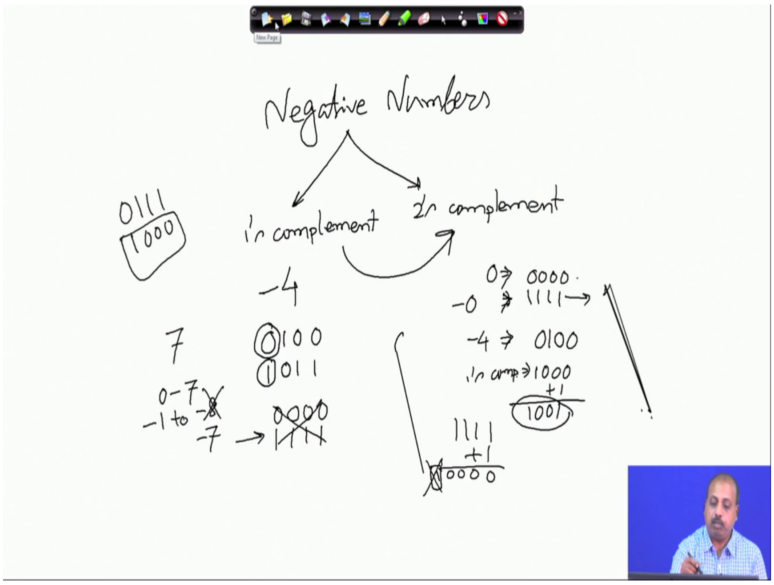
Microprocessors and Microcontrollers
Prof. Santanu Chattopadhyay
Department of E & EC Engineering
Indian Institute of Technology, Kharagpur

Lecture - 02
Introduction (Contd.)

So far whatever representation that we have seen, all the numbers they are positive in nature. So, we did not consider the negative numbers. But you see that negative numbers are quite common, like when you are doing particularly the subtraction operation so you can get a negative number and that negative number also has to be stored in the computer system.

(Refer Slide Time: 00:35)



For negative number storage, we have to think about some special representation. And when we are when we are talking about this binary number system, then this negative numbers are handled in 2 different ways. One representation is known as 1’s complement representation, 1’s complement representation and the other representation is known as 2’s complement representation.

So in 1’s complement representation, we take the complement of each of the bits to represent a negative number. For example, if I have the number say, if I have to represent the number minus 4, in a 4 bit number system then for a; first of all we take the representation of plus 4 in the 4 bit number system. So, plus 4 in 4 bit number system is represented like this 0100. So, if you oh represent, if you want to take represent minus 4 then, what will happen, we will complement each of the bits.

So, we start complementing, so, this is this bit becomes 1, this is 0, this is 1 and this is 1. So, this is the minus 4 representation in the 1's complement number system. So, this is very good because say I can represent numbers. So, if I have got say; so, this most significant bit of these numbers of this of any numbers. So, if this number be if the number is this bit is 0, so this represents positive number and if the most significant bit is 1 that represents negative number.

So, with that understanding, I have got 3 bits to represent the number. So, with 3 bits I can go from say, up to the number 7. So, if I have got say, 4 bit representation, I can represent I can represent the number 0 to 7 that are positive and I can take complements of them and I can represent minus 1 to, minus 1 to minus 8. Because, if I take the representation of 0 in 4 bit number system the number is 0, if I take ones complement, this will become all 1 and that is, sorry this is minus 0. So, this is wrong.

So, if I want to represent say plus 7, plus 7 is represented as 0111. Plus 7 is represented as 0111. So, if I want to represent minus 7, then it becomes 1000. Similarly, if I want to represent say minus 7, minus 7 is represented like this. So, this is not minus 8. So, this is going up to minus 7. So, I can go from 0 to plus 7 and minus 1 to minus 7. This is possible. However, there is a problem. Problem is that this 0, representation of 0, in a 4 bit number system 0 will be represented by this.

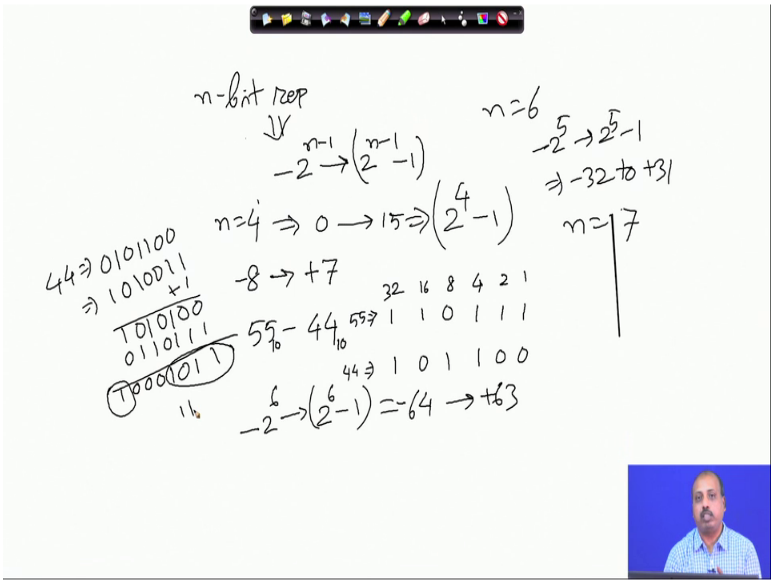
Now, if I complement all these bits, then what will happen; it will become 1111. That means, so what does it represent. As per our terminology, this should represent minus 0. So, this is the problem because now you have got 2 representations of the same number. Because plus 0 and minus 0 that is meaningless; they are same. But as far as the 1's complement representation is concerned, you have got 2 different representations for them.

So, that makes it difficult. So, because of this, we went for another representation which is known as 2's complement representation. In 2's complement representation, what is done; after getting 1's complement representation, we add 1 to it. So, minus 4 in 2's complement representation will be like this. So, to get the minus 4 in 2's complement, first we start with the 1s first we start with the representation of plus 4, 0100. Then we take the 1's complement, then we takes take the 1's complement which is 1000 and then we add a 1 with this. So that way it becomes 1001.

So, now you have got this number. So, this is the 2's complement representation of minus 4, fine. Now, this 2's complement representation, it does not suffer from this plus 0 minus 0 representations. Because, plus 0 is this 1. So, plus 0 is represented like this. Plus 0 is represented by all 0. Now if you have got minus 0, if you try to represent a minus 0, then what we will do is that we will take this minus complement of that so I will get this one.

Now, for 2's complement representation with this 1111, I will add with this 1111, I will add a 1. Then when I do this addition, all these bits will become 0 and with a carry of 1. So, if we discard this carry then it becomes 0. So, though both plus 0 and minus 0 they are represented by the same number which is all 0 in the number system. So, that makes the negative number representation easy.

(Refer Slide Time: 06:26)



So, there is, now there is a question of the range of numbers that you can represent. So, if you have got an n bit representation, if you have got an n bit representation, then this 2's complement number system, this will allow you to represent a sequence of numbers only. So, if you have if you have got n bit representation, n bit representation, then this 2's complement number system tells, you can represent minus 2 to the power n to minus 2 to the power n minus 1 to plus 2 to the power n minus 1 minus 1. So, you can represent this range of numbers. For example, if you have got n equal to 4; if you are do not consider negative numbers, then you can represent 0 to 15. That is, this 15 is nothing but 2 to the power 4 minus 1. So, we can go up to 0 to 15. But if you are representing both positive and negative numbers, then you can go from minus 2 to the power 3, that is minus 8 to 2 to the power n minus 1, that is 8 minus 1, 7. So, you can go from minus 8 to plus 7.

So, the rest of the numbers cannot be represented using a 2's complement number system in n bit. So, if you want to represent larger numbers, then you have to increase the value of entry, you have to represent it using more bits in the number system. So this way, we can do this thing. Now, how does it help us is that, this in 2's complement number system the addition and subtraction they become similar. So, how to do this; like if I have to do this subtraction say, 55 minus a 44, all these numbers are base 10 numbers, all these numbers are base 10 numbers.

So first of all, I have to take the binary representation of 55. So, if I, for binary representation it is easy if you just write down this chart on top 1, 2, 4, 8, 16 as the bits, as we are proceeding towards the left, the weight of the digits are increasing. So 55, this bit must be 1. So, 32 is gone. You are left with 23. So, this 16 must be equal to 1. So, 16, 23 minus 16, you are left with 7.

So, this is 48; 32 plus 16, 48. Now I cannot take 8 because that will make it 56. So, like I have to take this 4. So, that will make us 52 and both of this should be 1. So 11011, that is the representation of 55. So, this is 55; so for 44, again going by the same techniques 32 plus 8, 40, plus 1, 44.

Now, I want to represent minus 44; so, I must have one more digit. So, if I take say a representation of 6 bit, I cannot represent 44. Because 6 bit representation, I can go from minus; if you follow this formula, if n equal to 6. It is, it will, it can go up to minus 2 to the power 5 to 2 to the power 5 minus 1. So, the range is minus 32 to plus 31. So 44, minus 44 cannot be represented in this format.

So, I have to go for n equal to 7, I have to go for n equal to 7. If I go for n equal to 7, in that case I will be able to represent minus 2 to the power 6 to 2 to the power 6 minus 1. So, 2 to the power 6 is 64 to minus 64 to plus 63. So, these numbers fit in that range. So, I take a 7 bit representation, 6 bit representation sorry 7 bit representation.

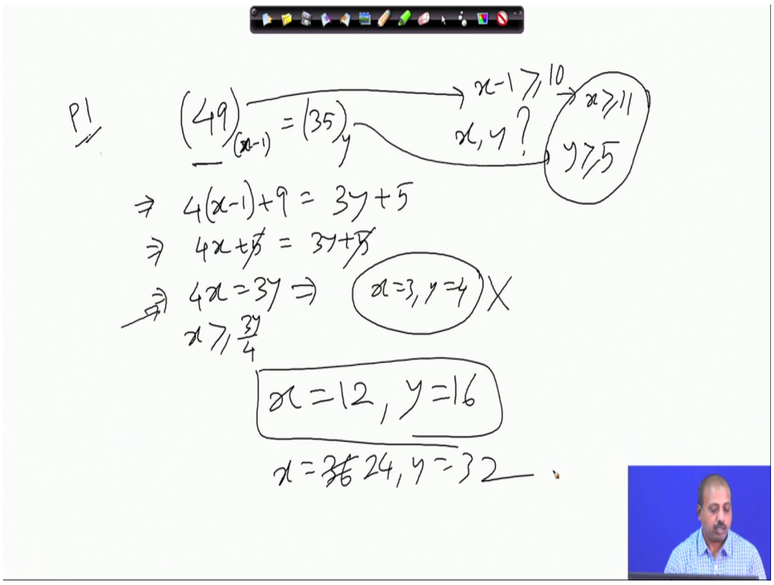
So, if I want to represent minus 44, then first of all I have to; so, plus 44, this is represented as 01, 0101100. So, I take first 1's complement. So that becomes 1010011. And then I add 1 with this. So, this becomes 0010101. Now, if I add this with 55. So, my operation is a subtract operation, but I do an addition.

So, addition how do I do? So, with this I add this number 55. So, 55 represented in 7 bit is like this, 0 first bit should be 0110111; do this addition. So, it is 110 then there is a carry of 1, then 0, then 1 comes, it becomes a 0, and there is a carry of 1. So, this number becomes this number becomes; so this bit I can ignore. So you see, this is nothing but 11 in the decimal number. In the decimal number system these value is 11, 8 plus 2 plus 1. So, this way I can get this addition operation done, addition and subtraction operation done in the same fashion.

So, this apparently it seems what is great in it! But the point is, when you are having some circuitry in your system, in your processor, then being able to do 2 operations done by a single piece of hardware is of much importance. Because then you do not have to have separate adder, separate subtractor whereas, for other number system like say 1's complement number system, you separate your add is subtraction modules should be separate. It cannot be clubbed with the addition module. So, that way it creates some difficulty.

Next, we will try to solve a few problems on this number system. Some class of some problem that we illustrate how these number systems are actually used in different cases.

(Refer Slide Time: 13:07)



Suppose we are, the first problem that will consider is same this one. It has got, we are given 2 numbers in 2 different number systems. The base values are not known, but it is said that the numbers are same then the values of the numbers are same then what will what can be the base value fine.

So, suppose the number given is the equation given is like this; 49 to the base x minus 1 is equal to 35 to the base y. So, this is the first number is a base x minus 1 number system and the second number is a base y number system and the values are same. So, we have to find out the values of x and y. What are the values of probable values of x and y. Of course, the answer may not be unique, but we can try to see what is happening.

So, if this has to be true then if we just apply our knowledge, of number system. So, 4 into x minus 1 plus 9 is equal to 3 into y plus 5. So, if you simplify it further, it gives 4 x plus 5 equal to 3 y plus 5. So, this 5 cancels out from 2 sides. So, we have got 4 x equal to 3 y. So, this is the requirement. How can this thing happen 4 x equal to 3 y; Now at the same time, we can have same this thing like this x should be greater or equal 3 y by four. So, that is the final condition that turns out to be coming here.

Now, this condition has to be satisfied. Now x, and there is another requirement, so there can be many values like this equation can be satisfied by putting x equal to 3 and y equal to 4. This equation can be satisfied. But can it be a solution? This x equal to 3, y equal to 4 can it be a solution to this system. But it is not because this x minus 1 is the base of this number system and in this base, what is the base now if x equal to 3? The base becomes equal to 2. So, the allowed digits are only 0 and 1, but here I am using the digits 4 and 9.

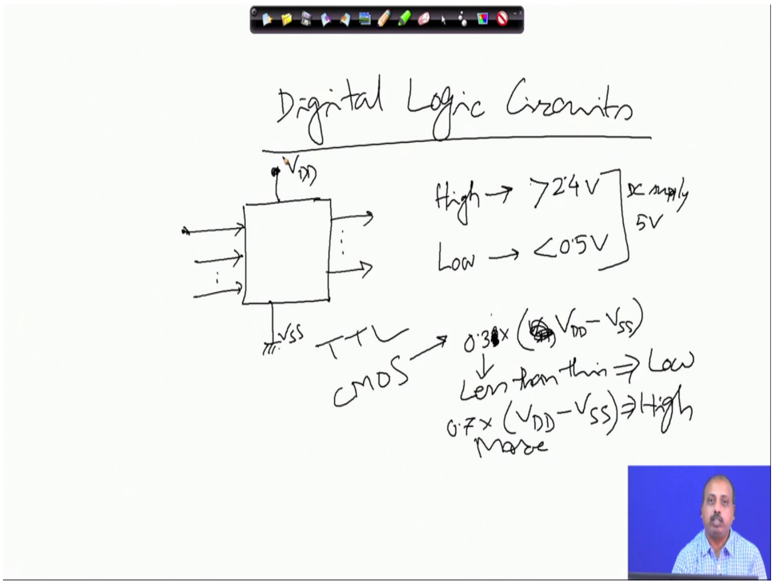
So, the x equal to 3 is not valid similarly, y equal to 4 is also not valid because here it is 35 so 5, the base we have got a digit which is 5 also base has to be more than 5. So, it cannot be equal to 4. So, this solution is incorrect solution. So, you cannot take this as the solution. So, another solution, from this equation I can say that x minus 1 should be greater than should be it should be greater or equal 10 because the 9 is there so the next digit is 10. So, from here I can say that x should be greater or equal 11. Similarly, from this I can say that y should be greater or equal 5. So, these are the additional constraint that we have over this solution. So, we have got 3 condition now; 4 x equal to 3 y, this is one condition; x greater or equal 11, y greater or equal 5, this is the other 2 constraints.

So, if you want to satisfy all these constraints, then what you get if as a as a probable solution is x equal to 12 and y equal to say 6 sorry not 6, y equal to 16. So, we have got this solution x equal to 12 and y equal to 16. So this way, if this solution, it satisfies this equation 4 x equal to 3 y. At the same time it satisfies these requirements x minus 1 greater or equal to 10 and y greater or equal 5, both the constraints are satisfied.

So, this is a, this is the correct solution. So, you can have other values of x as well. Like if you, if you multiply, if you if you go for next higher value. So, so say x equal to say 36, any multiple of this, x equal to the multiply both of them by 2 say, x equal to 24, y equal to 32. So, this is also a solution. So, this is there are numerous solutions, but it is one of those solutions ok.

So, next we will go to another part of, another part of this introduction lecture, which is about the basic digital design. So, we have seen the numbers they can be stored in the computer system or if the processor can process them using 2’s complement number system, but to for doing the processing, we need to have some circuitry. And as I have already said that, we will assume that information is available in the digital format.

(Refer Slide Time: 18:10)



So, operation that is done is in the digital format only. Now, when I am talking about this digital format data and their processing, we will be essentially talking about the digital logic circuits ok.

So, there can be, so this digital logic circuit, there are separate courses for this. So, we will not go into much detail, but the portions that we may need while doing our discussion on this microprocessor course. So, I will just try to touch upon those concepts. Now digital logic means, so, if this is a circuit which is taken as a digital block, it has got a number of inputs, it has got a number of inputs and there can be a number of outputs.

Now, unlike analog signal, where the values are continuous in nature; in digital system the values are discrete. And we have got 2 distinct discrete levels. So, one is called a high logic level sorry one is called a high logic level and the other one is called a low logic level. So, these high and low logic levels, they are basically 2 voltage levels that we consider, one is high another is low. Now there are several technologies that have been developed by in which these digital circuits are realized, 2 popular technologies are TTL and CMOS. So, these are the 2 popular technologies that we have.

Now, each of these logic, this logic families, they have got different interpretations of this high level and low level of signal. So, basically the signal that you are getting here is nothing but a voltage. So, interpretation of that voltage as logic high or 1 or logic low or 0, that depends on the level that the particular logic family considers ok.

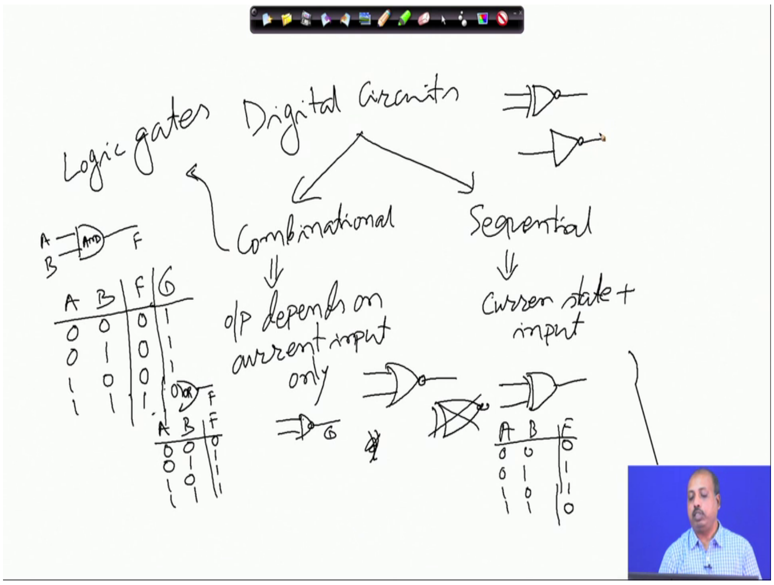
Now, in case of TTL this logic high is taken to be anything which is more than 2.4 volt, anything that is greater than 2.4 volt, that is taken as logic high and anything which is less than 0.5 volt, anything that is less than 0.5 volt is taken as logic low. On the other hand for CMOS, and here there the TTL family, the supply voltage, the supply voltage is DC supply voltage is 5 volt. So, with respect to that we have got this is a high logic level and low logic level. And this CMOS, it has got the cost consideration that is anything which is more than 0.33 sorry 0.3 into V DD, sorry 0.33 into V DD minus VSS.

So, that is taken as logic, anything that is more than, it is less than, this anything less than this is taken as logic low. So, less than this less than this is low, logic low. And anything which is more than 0.7 into this quantity, anything that is more than 0.7 into V DD minus V SS that is taken as logic high more than this quantity. Where this V DD, V SS, they are actually, if this is a circuit then it will have a supply voltage. So, in case of CMOS, the there are 2 point; one is called V DD and another is V SS. So, the in general this V SS point is the ground point and V DD is the supply voltage point.

So, that way it is similar to we have got say in case of TTL, we normally call it VCC, but in this case we will call it in V DD. So, this V DD and V SS, difference between these 2. So, the ground the V SS may not be equal to 0, V SS may have some other value. So, difference between V DD and V SS. So, that is taken as the difference, otherwise it is same. So, you see that there is a difference in the logic levels. And CMOS also has got various technologies, and in different technologies these V DD, V SS values they change. So, as a result we have got different as the technology scaling is progressing. So, now, the supply voltage is in the range of 1 volt.

So, naturally these high and low levels are much less compared to what we have in the TTL. So of course, each of these families they have got their own advantages and disadvantages. We will not go into that. And, both the families are used though for circuit implementation, it is mostly CMOS circuit. But, for the output stages of the circuits, many cases we go for TTL logic as well ok.

(Refer Slide Time: 23:30)



So, looking into the digital family, any Digital Circuit it can again be classified into 2 classes; one is called the Combinational circuit class combinational class and the other one is the Sequential class. So, combinational circuit means there is no memory. So, the circuit does not memorize it is previous state. So, wherever it was, from there it just produces the next; looking into the current input it will produce the output.

So, here output depends on current input only. Output depends on current input only. Whereas, in case of sequential circuits, output depends on current state of the system, current state plus input, it depends on both the quantities. So, it occurs as a Combinational circuit and Sequential circuit. Now for combinational circuit part, it is generally they are they are realized using something called Logic Gates and there are several types of Logic Gates that are that are there.

So, you may be familiar with the type of Logic Gate like AND. So, this is a, this is a symbol of an AND logic gate, a symbol of an AND logic gate. And if I have got 2 inputs, A and B and the output as is F, then it is represented in the form of a truth table. So, in the truth table we have got column corresponding to each of the prime, each of the inputs to the gate and a column corresponding to the output.

Now, we list down all possible combinations of these input values. So, if it is, they can be 0 0, 0 1, 1 0 and 1 1; so in general, will use 0 to represent the low value and 1 to represent the logic high value. Now, this AND gate says that, this output should be 0; this will also be 0; in this case also it should be 0; whenever both the inputs are 1, then only the output should be 1. So, this is the AND gate.

Similarly, there are many other gates like say, there is one gate called OR gate. So, which is represented like this, A and B are the 2 inputs and F is the output. So this is the OR gate. And here the truth table will be something like this. So, this is 0 0, 0 1, 1 0 and 1 1. And in case of OR gate, whenever any of the inputs is 1, output will be 1. So, this output will be 0, but in all other cases the output is equal to 1. Another, there are other gates like we have got NAND gate which is the invert of AND gate. So, which is the invert of AND.

So, in case of NAND, if I represent this by G, then this truth table will be modified to be something like G. So, all these will be 1 and when this was giving me a 0, whenever this was giving me a 0, 1. So in case of AND gate, here it will give me a 0. So, that is the NAND function. Similarly if you take the NOR gate; NOR gate is nothing but, this is the OR gate and there is an inversion at the top of it. So, this is the NOR gate.

Another very interesting gate is the XOR gate, where I have got again 2 input or they can be sorry this is not correct. So, we have got an XOR gate where say, suppose we have got 2 inputs and here the output will be 1 provided only one of them is equal to 1. So this is the truth table, if I draw this A, B and F. So, this is 0 0, 0 1, 1 0 and 1 1. So, when this F is, whenever, this is 0 because both the inputs are 0. This is one; this is one; but this is again 0, 1 1 is again 0 telling that the inputs are same. This XOR gate, it has got a very important application when we are going into this microprocessor designs like because you see it can compare between 2 inputs A and B. So, if you are trying to check whether 2 inputs are same or not, then this XOR gate is the straightaway the answer.

So, this is you can take the XOR of the 2 numbers or 2 inputs. And if the answer is 0, that means they are same. If the answer is 1, that means the values are not same. So, that way XOR gate has got very good application. So, other gates, similarly you can have XNOR gate as well where at the XOR gate output, you take a bubble. So, if this is the XOR gate, we take a bubble here, so that is the XNOR gate. And of course, another gate that we did not talk about explicitly is the inverter, which is a single input single output circuit. So, inverter means whatever be the input, the output will be the invert of that ok.