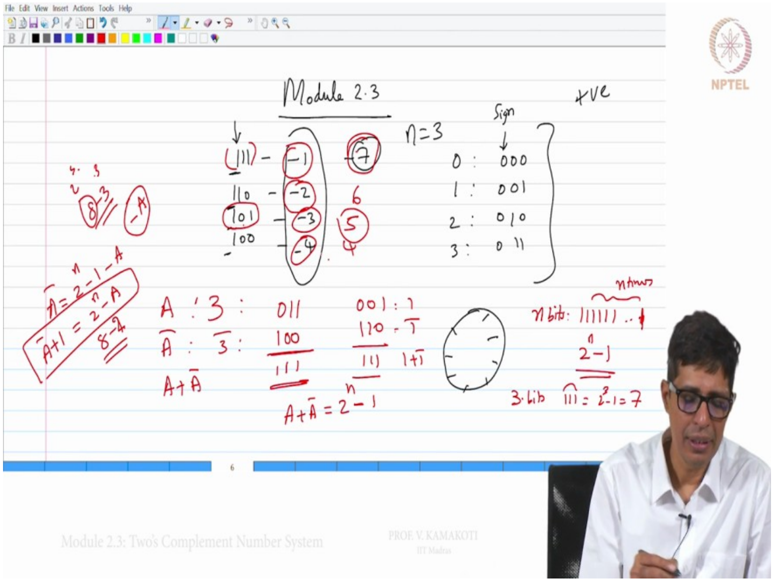
Foundations to Computer Systems Design.

Professor V. Kamakoti.
Department of Computer Science and Engineering.
Indian Institute of Technology, Madras.

Module 2.3.

Two's complement Number System.

(Refer Slide Time: 0:35)



So, welcome to module 2.3. We will now basically explain how a 2's complement number is generated. So, suppose there is a positive number, we just take the binary representation, normal binary representation. Let us again go back to n equal to 3, right. So if it is 0, it is, always 0 is treated as positive, so 000 this is a sign bit the sign bit, 1, 2, and 3, this is 001, 010, 011. So this is for the positive side. But then if you just look back into what we have done, 100, 101, 110, 111 and we made this -1, -2, -3, -4.

The interesting thing is that all the negative numbers you see here, the 1st bit is 1. So if the 1st bit is 1, you know that it is a negative number and this is how we have represented, right. They trusting, the other part is that since we are trying to map subtraction and addition together. The n equal to 3 cycle had basically 8 steps, so if I want to go one step backward, it is equivalent to 7 steps forward.

If I want to take 2 steps on the anticlockwise, sorry, one step on the anticlockwise sense, that is subtract -1, it is 7 steps forward or clockwise, 2 steps anticlockwise is 6 steps, 3 steps is 5 steps, 4 steps is 4 steps as again, right. And what you see here is this 7 is 111 in decimal, 6 is 110, 5 is 101 and 4 is 100. So if I say subtract for 4 or subtract 5, it means subtract 3, that is

-3, that means go 3 steps anticlockwise, that is equivalent to going 5 steps clockwise. So subtracting 3 means go that money steps of its representation clockwise.

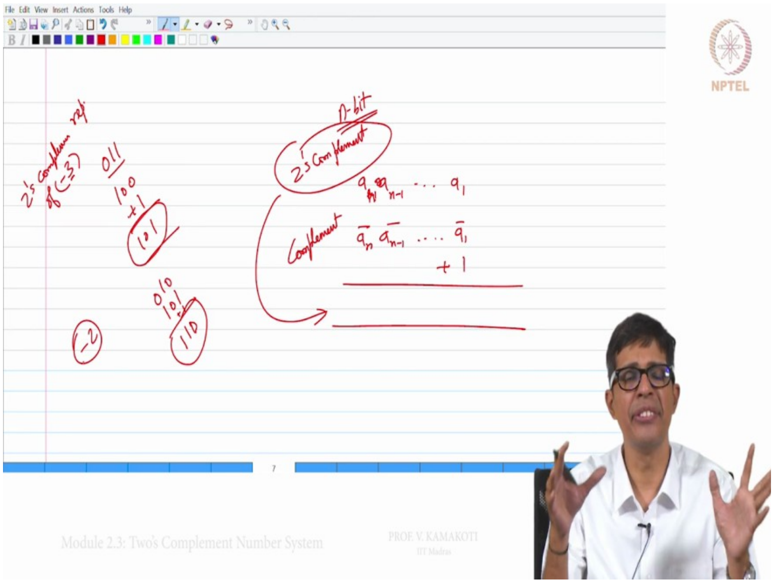
So subtracting 3 means move that many steps of its representation, what is its representation of -3, 101, so 101 is 5, so subtract 3 means go 5 steps clockwise, subtract 2 means go 6 steps clockwise. So this is how you are moving clockwise is basically addition, so this is how your subtraction actually becomes addition in 2's complement representation. The next thing that we need to learn here is given a number, how do you find its 2's complement representation?

For example, suppose, suppose I am given say 3, I want to find out -3. So given this number 3, say 011, suppose I complement it, complement of every bit, not of every bit, I get 100. So this is 3, this is 3 complement. So let me say this is A, this is A complement. Now, what is A + A complement, you will get 111, every time, any number. So let us take one, 001, 1's complement is 110. This is 111, these 1's complement, so 1+ 1's complement is again 111.

So what is 111 interesting? This is, so if I have n bits, 11111… n times is nothing but 2^{n-1}, right. For example, 3 bits, 111 is nothing but 2^{3-1} which is 7, right. So let A+ A's complement is 2^{n-1}, right. So A's complement is $^{2\ n\ -1\ -\ A}$. So if you add A's complement +1, this is 2^{n-A}. Right. So 2^{n-A} is nothing but A's complement +1. Now if I say subtract 3, that means I have to move on 8-3 steps clockwise. What is 8, 2^{3}, right.

So, what is the 2^{n-A}, it is the 2's complement representation of - A, right. For example what is 2^{n-1}, in this case 2^{38-1}, it is 7, 7 is the 2's complement representation of -1. Similarly it -2 is 6, 6 is the 2's complement representation of -2 and 8-3 is 5, 5 is a 2's complement representation of -3 and of course 4 is the 2's complement representation of -4, right.

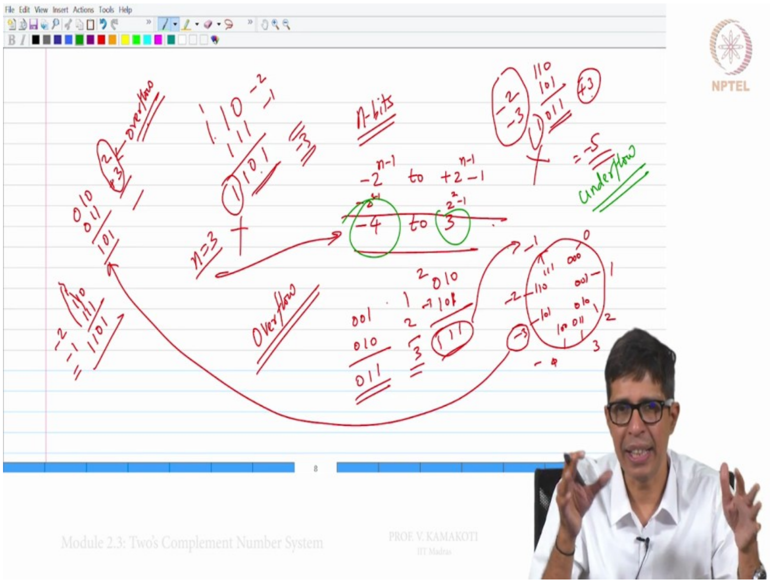
(Refer Slide Time: 6:26)



So, given any number, any n bit number, the way we get its 2's complement representation is as follows. Let this n bit number be a1, sorry, a n -1, sorry a n, a n -1 to a 1. You complement all the bits, so you get a 1 bar, a n -1 bar, a n bar and add 1 to it, so this would be the 2's complements representation. So just as an example, if I want 2's complement representation, 2's complement representation of same -3, take 3 first, that is 011, invert it 100, add 1, 101, note that the 2's complement representation of -3 is 101, right.

Similarly 2's complement representation of -2, -2, take 2, 2 is 010, 2's complement of that is, sorry complement of it will be 101, add 1 to this, it will become 110, 110 is the 2's complement representation of -2. So this is how we generate the 2's complement representation. So, all the integers that are basically stored inside the computer will be stored into is complement representation and any arithmetic, arithmetic in terms of subtraction or addition will be just, addition here. So, this add the 2 numbers given and you will get the annual get another number which will be the 2's complement representation, which will also be in its 2's complement representation.

(Refer Slide Time: 8:40)



So this is how basically the entire arithmetic works. So all the signed integer will be this 2's complement representation. Now we will just sum up this module by saying that suppose if I have n bits, I can represent between - -2^{n-1} to $+2^{n-1}-1$. For example, with n equal to 3, I can represent from -4 to, that is -2^{3-1}, -2 square 4 to +3, which is 2^{2-1}. So with the 3 bits I can represent from -4 to +3, right.

Now, one of the things that we need to keep in mind is overflow in the case of 2's complement representation . So let us go back to the circle that we did. So, this is 0, 1, 2, 3, -1, -2, -3, -4. Suppose I am adding 1+2, 001+010, I am getting 011, so 1+2 is 3. Suppose I am subtracting 2-3, so 2, -3 is 101, I get 111, what is 111, is -1. So this is 2, this is -3, the answer is -1, automatically you get 111, which automatically represents -1, we need not do any conversion here.

Suppose I am adding 2 +3, 010+011, right, 2 +3, so I get 101. What is 101 in this representation? Please note that 101 is -3, 101 is -3, so I am adding 2, I am adding 3, 2+, should have got 5 but in this representation 101 is -3, so essentially this addition has caused an overflow. Because, please note that 2+3 is 5 and 5 cannot be represented using 3 bits because using 3 bits I can only represent from -4 to +3. So I cannot represent 5 here. Automatically this lands up in a overflow.

Similarly if I do say -2 -1, -2 -1, so what is -2, 110, what is -1, 111, please add them, 0+1 is 1, 1+1 is 0, 1+1+1 is 11. 1+1+1 is 3 as you see here, okay, so I will again to it here. So 110+111, 1, 1+1 is 0, 1 goes as a carry, 1+1+1 is 1 and 1 is a carry. Since it is a 3 bit arithmetic, this carry goes off, we can only store up to 3 bits, this automatically goes away, it is not even stored. So, 101, what is 101, it is -3. So this is -2, -1, will automatically give you -3.

But what we see is -2-3, -2-3 is, -2 is 110, -3 is 101, add 1 1, 110, the carry is one, this carry anyway will go off. This will give me 011 which is +3. -2-3 should give me -5 but it is giving me +3, right, because 011 represents +3, this carry will go off because this computer can represent only 3 bits, so that 1 will not even come into the picture. So why it is so? This is a wrong result as far as we are concerned, 011 is +3 in a representation, which is wrong. And that has happened because -2 - is going to give me -5 and that -5 is not within the range of -4 to 3.

So this is also sometimes, some books called it overflow, some books quality underflow because we are going below the negative value. I am trying to get a number which is lesser than this range, -5 is lesser than -4, while on the other hand I got a number which is greater than this range, right. So this is something that we need to understand in terms of 2's complement representation.

So with this in the last 2 modules I have covered 2's complement representation, wherein binary subtraction becomes binary addition. We have also got a representation for signed integers, which we will be using in this course. So I hope all of you understood, if you have doubts, you can put it on the forum, we will try and explain. But there are n Wikipedia and other sources, again which explains this 2's complement.

Whatever I have told is not so clear to you, you can go for a more elaborate treatment of this in Wikipedia. But any engineer, not just computer scientist or electrical or electronic person, every engineer today should understand how integers are indeed represented inside a computer. If you do not have that knowledge, you cannot do anything with computers. So kindly spend sometime in understanding and also try and put little more in the assignment, wherein we can, we will test your understanding of this 2's complement arithmetic, this is very very important. Thank you very much.