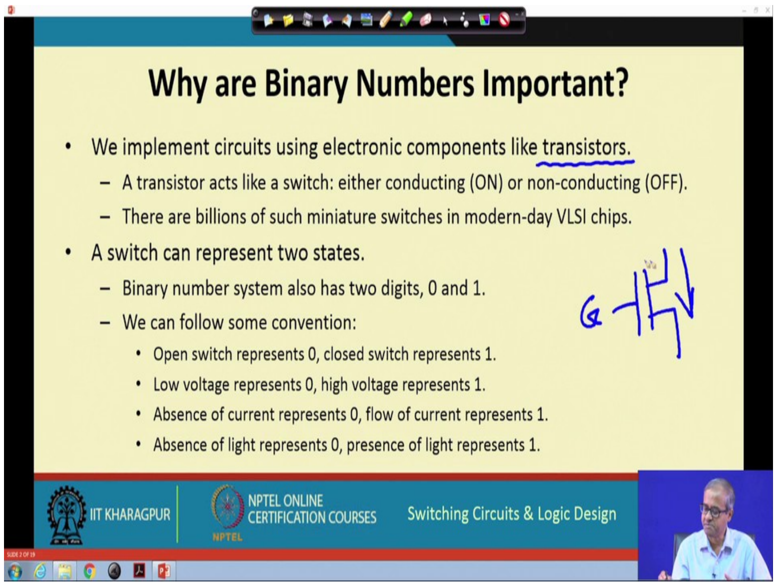
Switching Circuits and Logic Design

Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 03
Signed and Unsigned Binary Number Representation

So, if you recall in the previous lectures, we talked about the various number systems decimal, binary, octal, hexadecimal. And we mention one thing that binary number system is important from the point of view of circuit design, specifically digital circuit design. So, in this lecture we shall try to first motivate you why that is so, and secondly we shall show you or tell you that how to represent arbitrary numbers in binary like. So, far we did not considered negative numbers how do we represent negative numbers, we assumed all numbers to be positive ok. So, we shall be talking about these specific things in the current lecture. So, this lecture is titled Signed and Unsigned Binary Number Representation.

(Refer Slide Time: 01:13)

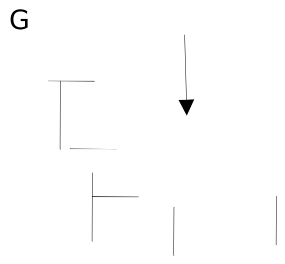


Why are Binary Numbers Important?

- We implement circuits using electronic components like transistors.
  - A transistor acts like a switch: either conducting (ON) or non-conducting (OFF).
  - There are billions of such miniature switches in modern-day VLSI chips.
- A switch can represent two states.
  - Binary number system also has two digits, 0 and 1.
  - We can follow some convention:
    - Open switch represents 0, closed switch represents 1.
    - Low voltage represents 0, high voltage represents 1.
    - Absence of current represents 0, flow of current represents 1.
    - Absence of light represents 0, presence of light represents 1.

G
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Switching Circuits & Logic Design

So, let us first motivate ourselves why binary numbers are important, why do we use them. So, I just mentioned that binary numbers are important from the point of view of your circuit design, but why is it so ok. To answer this question you will have to understand how circuits are designed and implemented, modern day circuits are invariably implemented using basic building blocks like transistors, they are typically MOS transistors, metal oxide semiconductor symbolically a transistor is represented like this.



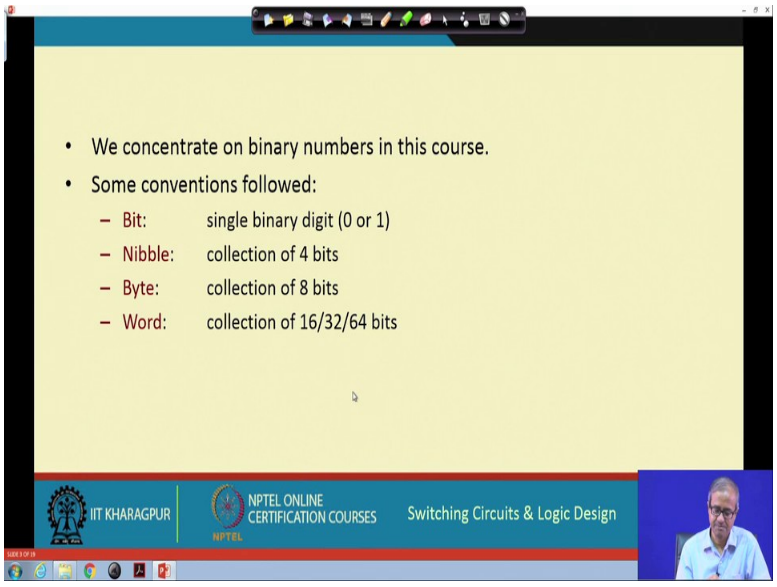
Depending on what we apply to a terminal called gate, there may be a current flowing between the other 2 terminals or there may not be any current flowing. So, transistor acts like a switch depending on my control input G. G can be regarded as a controlling input. So, it is either conducting or non-conducting, there are 2 states. Now, in a modern day VLSI chip, there are billions of such transistors; that means, there are billions of such switches very small in size, they all turn on and turn off in synchronism depending on some external event or applied input and, they realize some functionality that we wanted to realize ok.

So, as it said a switch can represent 2 states either conducting or non-conducting. Now, as an analogy binary number system also has 2 states or digit 0 and 1. So, if I can represent the conducting state as 1 and the non conducting state as 0, then I can say that a MOS transistor the state of a MOS transistor represents a binary digit. It is either conducting or non conducting means it is either 1 or 0. So, there is a one to one correspondence.

Similarly, other conventions you can follow, this is what I just mention open switch is 0 close switch is 1, or you can talk about voltages at some point if the voltage is low you can say it is 0, if the voltage is high you can say it represents binary 1. Then absence of current or presence of current like the example I took, or in some circuits nowadays we use light for the purpose of computation.

So, if there is no light it can represent binary 0, if there is a light it can represent binary 1. This is the basic principle behind photonic communication like, you know nowadays optical fibers are used for long distance high speed communication the basic principle is like this, you send light and you sense light at the other side whether the light is coming or not coming. So, bits can be represented depending on the presence or absence of light ok.

(Refer Slide Time: 04:45)

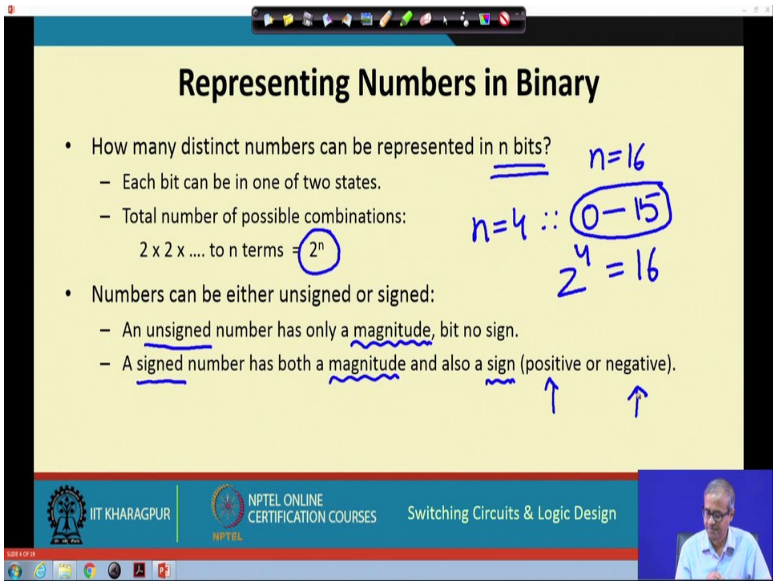


- We concentrate on binary numbers in this course.
- Some conventions followed:
  - Bit: single binary digit (0 or 1)
  - Nibble: collection of 4 bits
  - Byte: collection of 8 bits
  - Word: collection of 16/32/64 bits

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Switching Circuits & Logic Design

So, in this course because we are talking about digital circuits, we shall be concentrating on binary numbers. Now, let us introduce some definitions, a single binary digit which is 0 or 1 is termed as a bit as you also mentioned in the last lecture. Now, when you have a collection of 4 bits that is called a nibble, collection of 8 bits is called a byte. Next higher thing we call it as a word, but of course, the definition word is not well defined, depending on the context 16 bits, 32 bits or 64 bits can be considered as a word, but the definition of bit, nibble and byte are well defined. It is 1 4 and 8 that many bits right.

(Refer Slide Time: 05:44)



Representing Numbers in Binary

- How many distinct numbers can be represented in n bits?
  - Each bit can be in one of two states.
  - Total number of possible combinations:

    2 x 2 x .... to n terms = $\boxed{2^n}$
- Numbers can be either unsigned or signed:
  - An unsigned number has only a magnitude, bit no sign.
  - A signed number has both a magnitude and also a sign (positive or negative).

$n=16$

$n=4 :: \boxed{0-15}$

$2^4 = 16$

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Switching Circuits & Logic Design

Now let us talk about representing numbers in binary. So, the question that we are trying to answer is that. Suppose we are representing a binary number in n number of bits, let us say n equal to 16 or something let us say n equal to 16.

Now, I want to ask a question that in 16 bits, how many distinct numbers can I represent. This of course, I want to know like in 4 bits you had already seen earlier that when you are talked about hexadecimal numbers, in 4 bits we can represents number from 0 up to 15, octal for 3 bits 0 up to 7. So, my question is for n bits in general, what will be the range of numbers how many total numbers I can represent ok. So, it is a simple combination calculation at each bit position there are 16 bit positions right. So, each bit positions can be either 0 or 1.

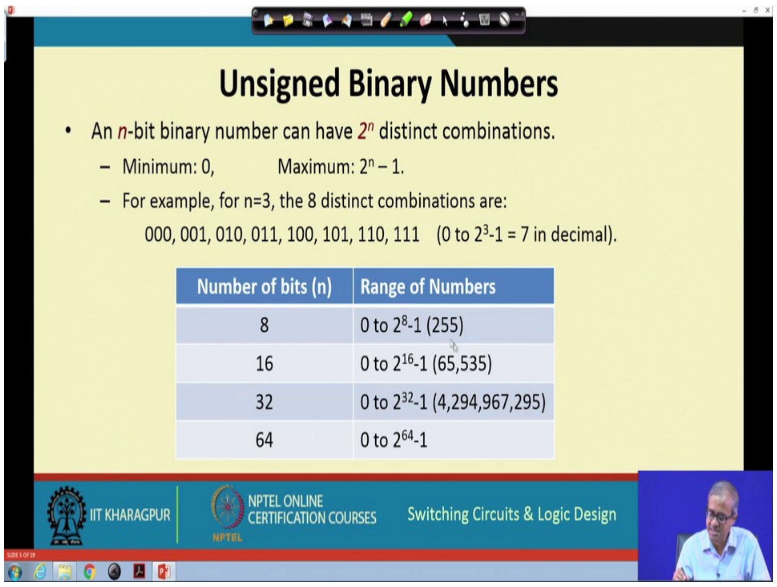
So, there are 2 combinations, second bit position again 0 or 1. So, you multiply 2 into 2 into 2 up to n times. So, you get 2 to the power n this is the total distinct numbers you can represent. So, so you recall for hexadecimal n equal to 4 you had numbers in the range of 0 to 15.

$$n \times ¿ = 2^n$$
$$2 \times 2 \times \ldots \times 2 ¿$$

So, how many numbers are possible 2 to the power 4, 2 to the power 4 means 16. So, total 16 numbers could be represented. So, here for n bits I can represent total of 2 to the power n numbers ok. Now, next question is how to incorporate sign in a number numbers can be unsigned, or there can also be a sign associated with the number. So, unsigned number will only have a magnitude, but no sign, but signed number will be having a magnitude as well as a sign, which will indicate that it is either positive or it is negative right ok.

$$Total\,number\;can\,be\,represented = 2^n \quad , \text{f} \quad ¿ n = 4, 2^4 = 16$$

(Refer Slide Time: 08:11)



Unsigned Binary Numbers

- An *n*-bit binary number can have 2^n distinct combinations.
  - Minimum: 0, Maximum: $2^n - 1$.
  - For example, for n=3, the 8 distinct combinations are:

    000, 001, 010, 011, 100, 101, 110, 111 (0 to 2^3-1 = 7 in decimal).

Number of bits (n)	Range of Numbers
8	0 to 2^8-1 (255)
16	0 to 2^{16}-1 (65,535)
32	0 to 2^{32}-1 (4,294,967,295)
64	0 to 2^{64}-1

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Switching Circuits & Logic Design

Let us first talk about unsigned binary numbers. Now, we just now saw that in n bits you can have 2 to the power n distinct combinations. So, the minimum number that you can represent is 0 all 0's and maximum number is 2 to the power n minus 1 all 1. You think of hexadecimal 000 means 0 and 111 means 15. For octal if you take n equal to 3 the 8 distinct combinations should be 000, 001 up to 111, if you count there are 8 combinations 000 means 0, 111 means 7 which is 2 to the power 3 minus 1, 2 to the n minus 1 ok.

$$Minimum\, number = 00\ldots0\,(n\, zeros) = 0$$

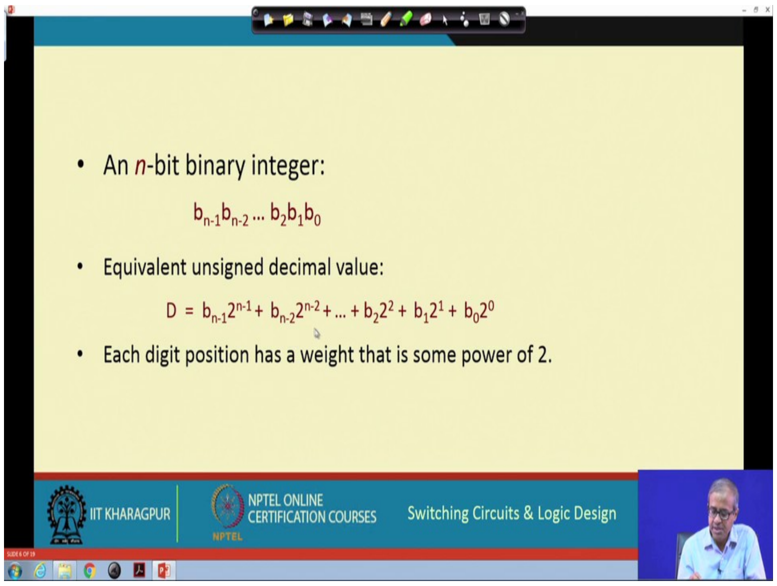
$$Maximum\, number = 11\ldots1\,|n\, ones| = 2^n - 1$$

This table shows that as the value of n increases. So, how much is the range of the numbers change for 8 bits, it will go up to 2 to the power 8 minus 1 means 256 minus 1, which means 255, for 16 bits this is 65 535 for 32 bits it will be about 4 billion. So, you can say numbers up to 10 digits can be represent, but in 64 it will be even larger. So, if you just calculate this it will be approximately about 21 digits. So, large number so, as the number of bits increases the range of the number increases very rapidly ok. So, as this table shows, this you have to remember.

$$For\, n=8, the\, range\, is\, 0 ¿ 2^8 - 1, i.e.0 ¿ 255$$

$$For\, n=16, the\, range\, is\, 0 ¿ 2^{16} - 1, i.e.0 ¿ 65{,}536$$

(Refer Slide Time: 10:00)


- An *n*-bit binary integer:

  $b_{n-1}b_{n-2} \dots b_2b_1b_0$

- Equivalent unsigned decimal value:

  $D = b_{n-1}2^{n-1} + b_{n-2}2^{n-2} + \dots + b_22^2 + b_12^1 + b_02^0$

- Each digit position has a weight that is some power of 2.

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NPTEL
Switching Circuits & Logic Design

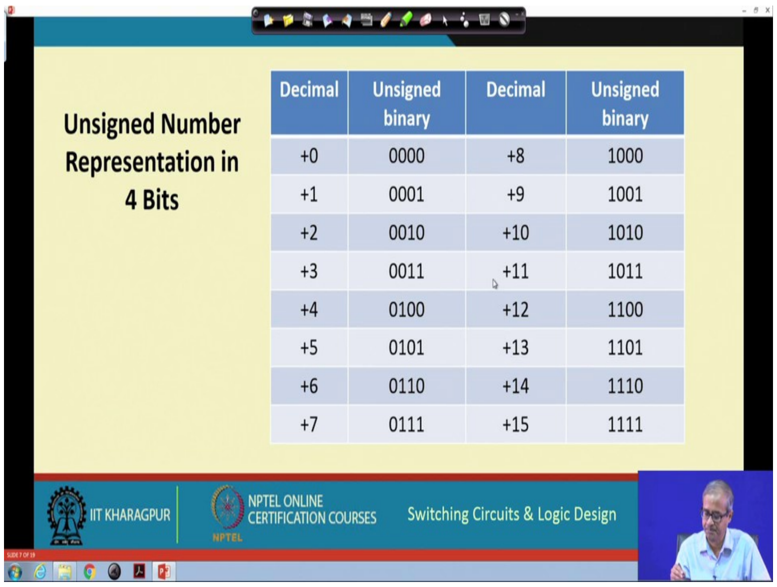
Now, this already we have to talked about earlier, this I am showing it in a, you can say in a mathematical form. Let us say I have an n bit binary number, where the digits are represented as this is an integer number no fractions point b0 b1 b2 up to bn minus 1. b0 is the least significant bit position and bn minus 1 is the most significant bit position.

$$An\, n\, bit\, binary\, number = b_{n-1}b_{n-2}\dots b_2 b_1 b_0$$

So, the weights will be 2 to the power 0 (2^0), 2 to the power 1 (2^1), 2 to the power 2 (2^2) and so, on. So, when you try to calculate the decimal equivalent, you multiply the digits by the corresponding weights and add them up, like b0 you multiply by 2 to the power 0 (2^0), b1 by 2 to the power 1 (2^1), and so on bn minus 1 by 2 to the power n minus 1 (2^{n-1}). So, in binary we have seen that each digit position has a weight, which is some power of 2, this way you can calculate the decimal equivalent ok.

$$The\, decimal\, equivalent = b_{n-1}\times 2^{n-1} + \dots + b_2\times 2^2 + b_1\times 2^1 + b_0\times 2^0$$

(Refer Slide Time: 11:09)



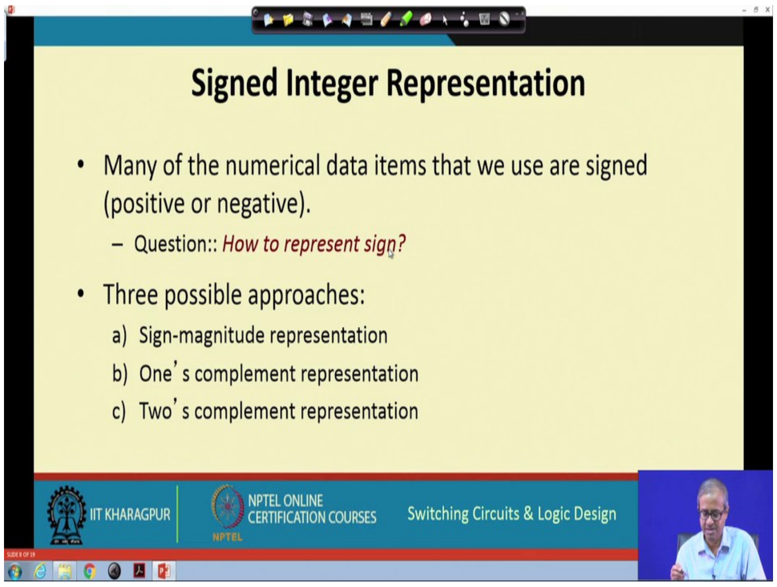
Unsigned Number Representation in 4 Bits

Decimal	Unsigned binary	Decimal	Unsigned binary
+0	0000	+8	1000
+1	0001	+9	1001
+2	0010	+10	1010
+3	0011	+11	1011
+4	0100	+12	1100
+5	0101	+13	1101
+6	0110	+14	1110
+7	0111	+15	1111

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Switching Circuits & Logic Design

Just an example for n equal to 4 for 4 digit just like hexadecimal so, if you express all possible combinations in binary and try to convert them to decimal, you will see that the range of numbers you can represent will start with plus 0, it will go up to plus 15. These are considered to be unsigned number; that means, positive no sign ok. This already we have seen for hexadecimal numbers. Same thing is there.

(Refer Slide Time: 11:45)



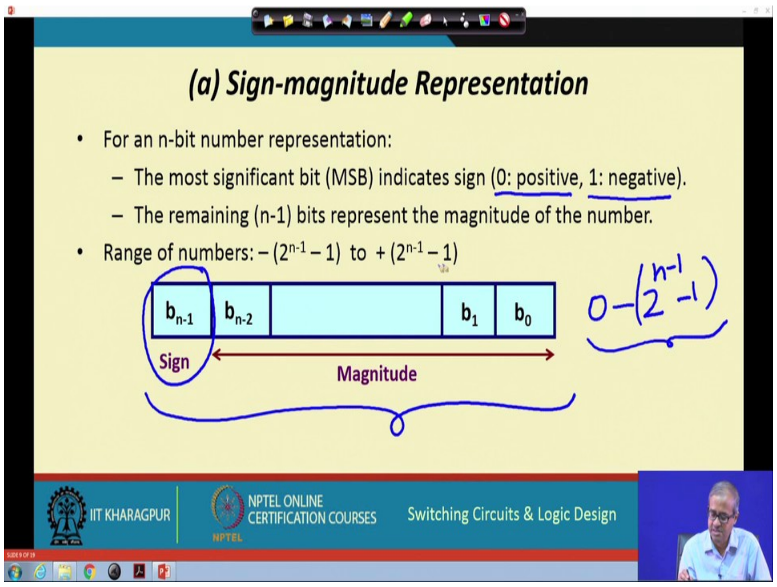
Signed Integer Representation

- Many of the numerical data items that we use are signed (positive or negative).
  - Question:: *How to represent sign?*
- Three possible approaches:
  a) Sign-magnitude representation
  b) One's complement representation
  c) Two's complement representation

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Switching Circuits & Logic Design

Now let us come to the issue of sign. So, in a most of our calculations we have to deal with both positive and negative integers. So, now, the question is how do you represent negative numbers in a system or in a circuit and how do we manipulate on them ok. So, the answer to this question how to represent sign there are 3 broad approaches which are used in practice, these are called sign magnitude one's complement and two's complement representations. Now, we shall be explaining these three representations one by one. These are all used to represent signed numbers, numbers which are both positive and negative.

(Refer Slide Time: 12:38)



(a) Sign-magnitude Representation

- For an n-bit number representation:
  - The most significant bit (MSB) indicates sign (0: positive, 1: negative).
  - The remaining (n-1) bits represent the magnitude of the number.
- Range of numbers: $-(2^{n-1}-1)$ to $+(2^{n-1}-1)$

Let us look at the simplest first the sign magnitude representation. You see this sign magnitude representation conceptually looks like this. Suppose I have an n bit representation this whole thing is n bits. So, I reserve 1 bit for this sign, this is my sign the remaining n minus 1 bits will represent magnitude.

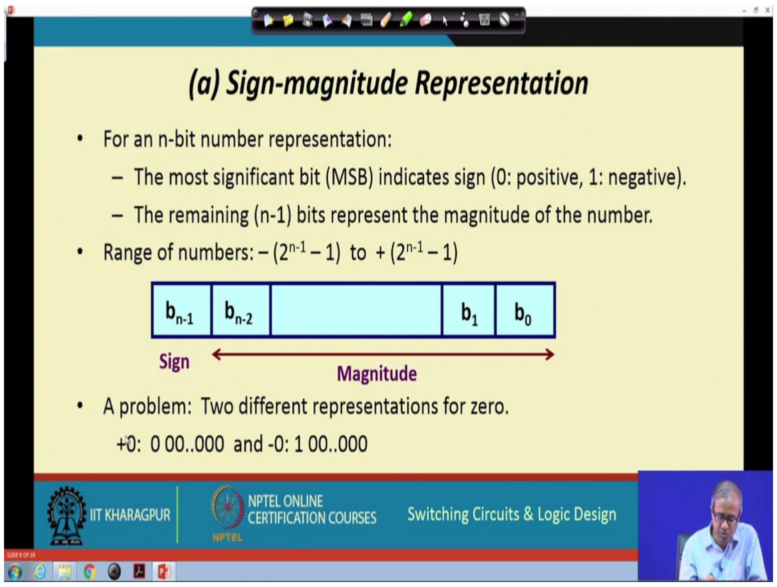
So, as a matter of convention we can follow this in this sign, 0 means the number is positive, 1 means the number is negative. So, the most significant bit or MSB, in short it is called it represents the sign, which can be 0 or 1. 0 is positive, 1 is negative. And the remaining n minus 1 bits will represent the magnitude. And already we know that in n minus 1 we can represent a number in the range 0 up to 2 to the power n minus 1 minus 1 ($2^{n-1}-1$). So, the magnitude can be in this range.

$$For\, n-1\, bit, the\, range\, is\, 0 ¿ (2^{n-1}-1)$$

So, in this representation the range of numbers can be, sign will indicate whether its positive or negative, but the maximum magnitude can be 2 to the power n minus 1 minus 1 (

$2^{n-1}-1$). So, this smallest number will be this, the largest number will be this right. This follows directly from the binary representation.

(Refer Slide Time: 14:20)



(a) Sign-magnitude Representation

- For an n-bit number representation:
  - The most significant bit (MSB) indicates sign (0: positive, 1: negative).
  - The remaining (n-1) bits represent the magnitude of the number.
- Range of numbers: $-(2^{n-1}-1)$ to $+(2^{n-1}-1)$

b_{n-1}
b_{n-2}
b_1
b_0
Sign
Magnitude

- A problem: Two different representations for zero.

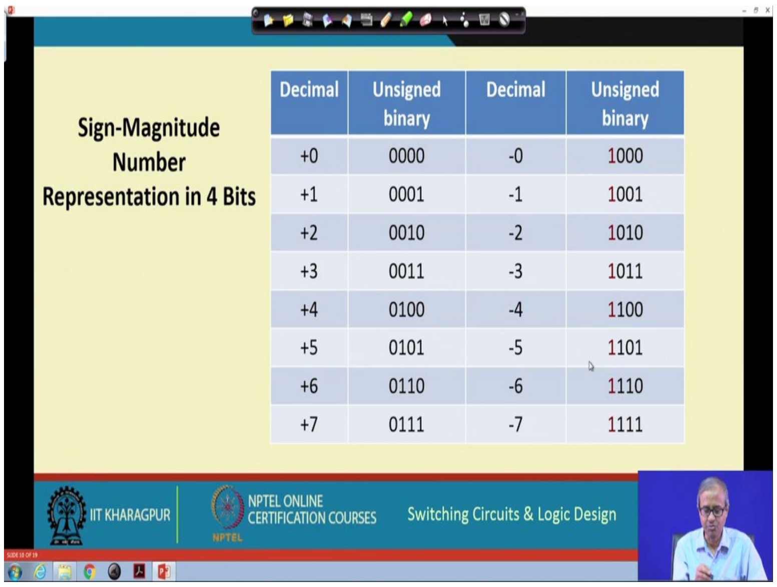
  +0: 0 00..000 and -0: 1 00..000

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Switching Circuits & Logic Design

But this one issue here, you can see that in this representation 0 is represented by 2 different bit patterns because, for the number 0 the magnitude obviously will be 0. So, this will be all 0's, but this sign can be either 0 or 1, it does not matter because 1 will indicate plus 0, other will indicate minus 0. But technically speaking both are 0's right plus 0 or minus 0.

So, one drawback of this representation is that 1 bit pattern we are wasting, we are using up two different bit patterns to represent the same number 0 ok. This is one small drawback in this representation.

(Refer Slide Time: 15:08)



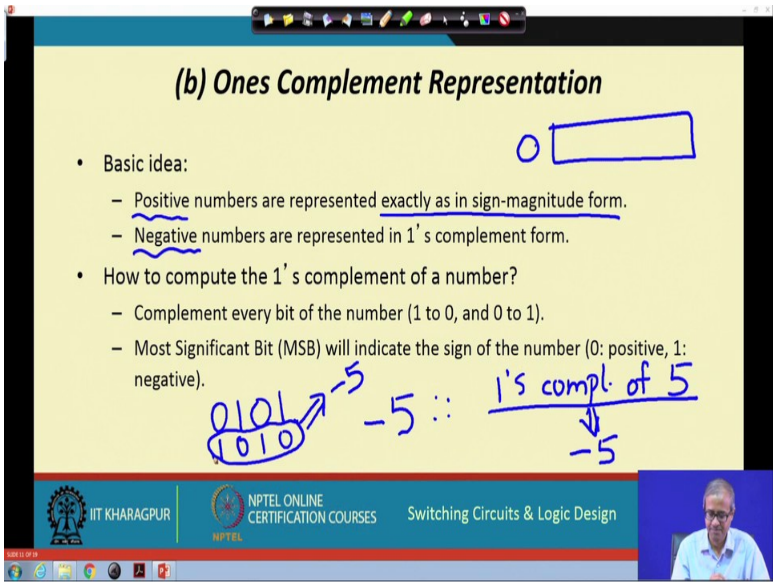
Sign-Magnitude Number Representation in 4 Bits

Decimal	Unsigned binary	Decimal	Unsigned binary
+0	0000	-0	1000
+1	0001	-1	1001
+2	0010	-2	1010
+3	0011	-3	1011
+4	0100	-4	1100
+5	0101	-5	1101
+6	0110	-6	1110
+7	0111	-7	1111

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Switching Circuits & Logic Design

So, this table shows you the sign magnitude representation in 4 bits, if it is 4 bits you see the first column represents the MSB most significant bit is 0, which means the numbers are positive. And the last 3 bits indicate magnitude 000 is 0, 011 is 3. So, on 110 is 6, 111 is 7 and second column indicates, where the most significant bit is 1, 1 indicates the number is negative. But the magnitude is again ranging from 0 to 7. So, you see that you can represent a total of 15 distinct numbers because, plus 0 and minus 0 are having 2 different representations and on one side I can represent plus 1 to plus 7, on the other side minus 1 to minus 7. So, 14 plus 1, 15 total numbers I can represent ok.

(Refer Slide Time: 16:11)



(b) Ones Complement Representation

- Basic idea:
  - Positive numbers are represented exactly as in sign-magnitude form.
  - Negative numbers are represented in 1's complement form.
- How to compute the 1's complement of a number?
  - Complement every bit of the number (1 to 0, and 0 to 1).
  - Most Significant Bit (MSB) will indicate the sign of the number (0: positive, 1: negative).

0
0101
1010
-5
-5 ::
1's compl. of 5
-5
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Switching Circuits & Logic Design

Let us come to 1's complement representation. So, what is 1's complement representation, in this method if the number is positive, then it is represented exactly like in sign magnitude form no difference. The first bit indicates sign which is 0 and the remaining part represents magnitude, but negative numbers are represented in a different way that we should see. Suppose I have a number minus 5 so, I do something called 1's complement of 5, 1's complement of 5.

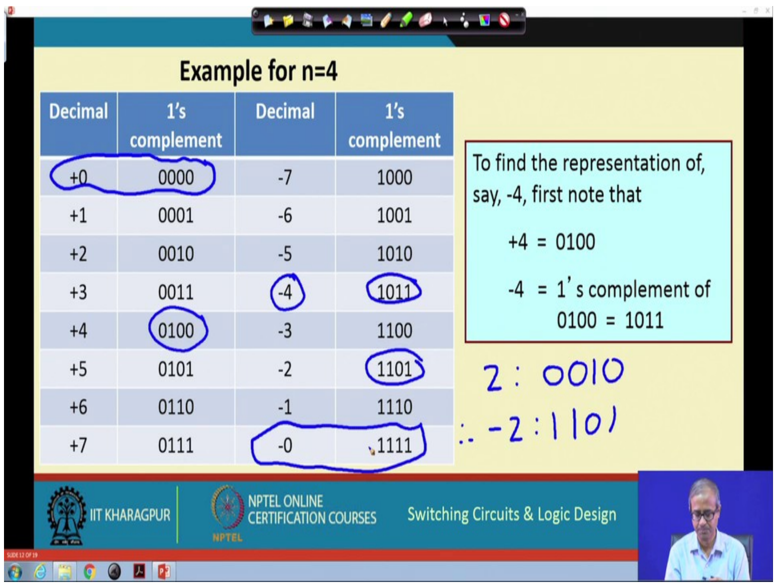
So, it is explained what is 1's complement. So, once I do ones complement of 5, this is my representation for minus 5. So, when I have a negative number I take the magnitude 5, I compute something called 1's complement of 5 and that will give me my representation of minus 5. So, what is ones complement it is very simple, you take the number let us say we have 5, let us say in 4 bits 5 is what 0101 this is 5 right, in 4 bits. You compliment every bit of a number. That means, 1 you make, 0 you make 1, 1 you make 0, 0 you make 1, 1 you make 0. You change or flip every number and this will be your 1's complement. This 1010 will indicate minus 5 right.

$$4\, bit\, representation, +5=0101$$

$$1'scomplement\, of +5=1010=-5$$

Now, you see here again the most significant bit will indicate sign because, when you do this the plus 5 was having the most significant bit as 0, but as you flip it has become 1. So, for negative numbers the MSB will automatically be still 1. So, just by looking at the most significant bit, you can tell whether the number is positive, or whether the number was negative ok.

(Refer Slide Time: 18:41)



Example for n=4

Decimal	1's complement	Decimal	1's complement
+0	0000	-7	1000
+1	0001	-6	1001
+2	0010	-5	1010
+3	0011	-4	1011
+4	0100	-3	1100
+5	0101	-2	1101
+6	0110	-1	1110
+7	0111	-0	1111

To find the representation of, say, -4, first note that

+4 = 0100

-4 = 1's complement of 0100 = 1011

2 : 0010

∴ -2 : 1101

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Switching Circuits & Logic Design

So, this table shows you the representations in 1’s complement form. For the positive numbers as I said they are same as the sign magnitude form 0 1 2 3 4 5 6 7, but for negative numbers it is a little different. Well here, I worked out one. Let us say I want to represent minus 4, I want to represent minus 4. So, how do I do it. I first take plus 4, what is plus 4, plus 4 is my 0100, I take 0100, then I take 1’s complement of 0100. I flip the bits 0 becomes 1, 1 becomes 0, they become 1 and 1. So, it is 1011, this way you can take something, let us take minus 2, 2 means 0010.

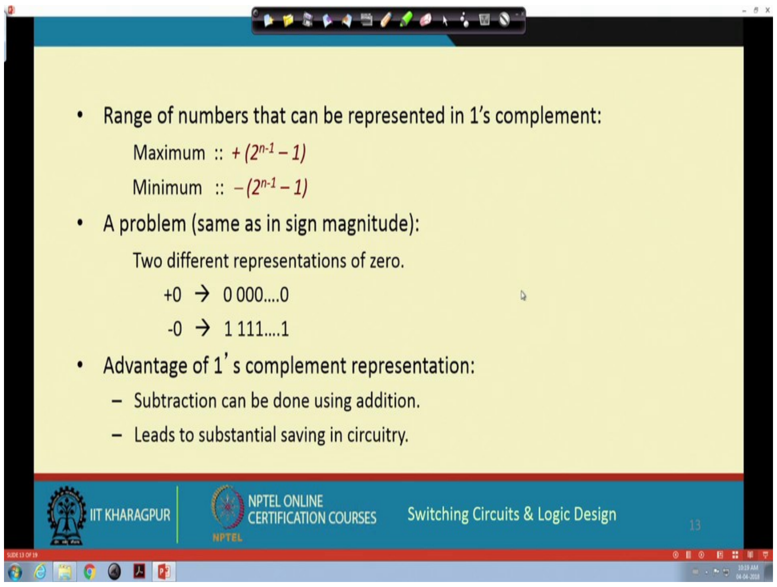
$$+4=0100, 1' s complement\, of +4=1011=-4$$

$$+2=0010, 1' s complement\, of +2=1101=-2$$

So, therefore, minus 2 will be take 1’s complement 1101, you see minus 2 is 1101. So, in this way, we can calculate for everyone. So, here again you will see that there will be 2 distinct representations coming from 0, plus 0 will be having representation 0000, minus 0 will be having representation 111, because you take plus 0 make negative 1111, both represents 0. So, here again 1 of the combination is wasted just like sign magnitude.

$+0=0000, 1'scomplement\,of +0=1111=-0$

(Refer Slide Time: 20:32)



- Range of numbers that can be represented in 1's complement:
  - Maximum :: $+(2^{n-1}-1)$
  - Minimum :: $-(2^{n-1}-1)$
- A problem (same as in sign magnitude):
  - Two different representations of zero.
    - +0 → 0 000....0
    - -0 → 1 111....1
- Advantage of 1's complement representation:
  - Subtraction can be done using addition.
  - Leads to substantial saving in circuitry.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | Switching Circuits & Logic Design 13

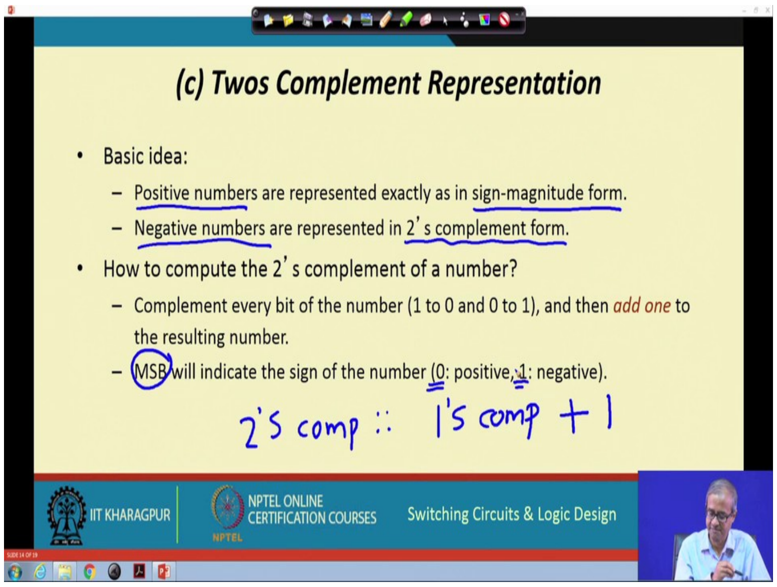
So, it is mentioned here the range of numbers that can represented here is just like sign magnitude form, maximum is plus 2 to the power n minus 1 minus 1 [$\overset{2}{\underset{+¿}{(¿¿n-1-1)}}$, minimum is minus 2 to the power n minus 1 minus 1 [$\overset{2}{\underset{-¿}{(¿¿n-1-1)}}$. Because, we look in the previous one, in the positive side it was plus 7, negative side also it was minus 7 ok. So, plus for n equal to 4 it becomes 2 to the power 3 minus 1 7, minus 7. And also this I mentioned that there will be 2 different representations of 0.

$$Maximum\,number=+\left(2^{n-1}-1\right)$$

$$Minimum\,number=-\left(2^{n-1}-1\right)$$

But one big advantages that this of course, we shall not be discussing now we will discussing later, that one big advantage of 1's complement representation is that, when you do subtraction you do not need to separately subtract numbers because, you can subtract numbers using addition only. So, what does it mean, well in a computer system we have both addition circuits, we are both subtraction circuits. Now, we are saying that if we represent our numbers in 1's complement form, we do not need subtractor circuits. We can have only the adder circuit using adder, we can do addition, you can also do subtraction, this is a big advantage. So, we shall see later how this is, how this is done or how this is achieved ok.

(Refer Slide Time: 22:11)



(c) Twos Complement Representation

- Basic idea:
  - Positive numbers are represented exactly as in sign-magnitude form.
  - Negative numbers are represented in 2's complement form.
- How to compute the 2's complement of a number?
  - Complement every bit of the number (1 to 0 and 0 to 1), and then *add one* to the resulting number.
  - MSB will indicate the sign of the number (0: positive, 1: negative).

2's comp :: 1's comp + 1

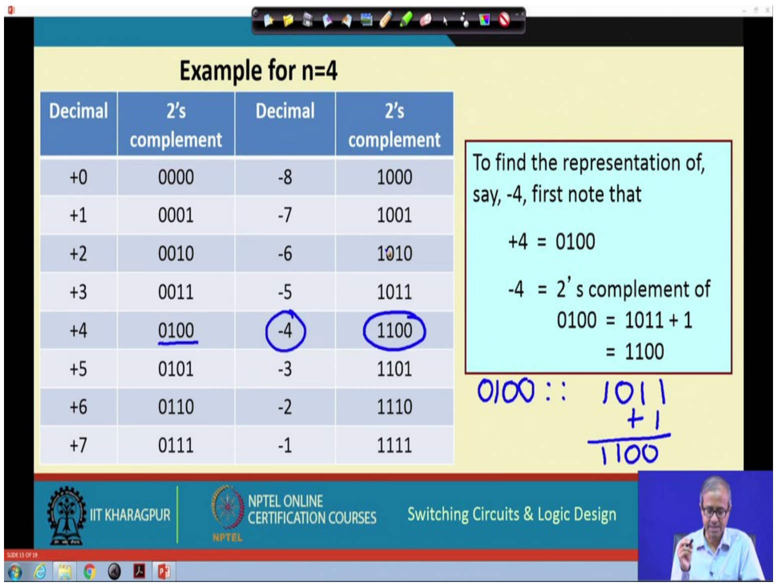
IIT KHARAGPUR NPTEL ONLINE CERTIFICATION COURSES Switching Circuits & Logic Design

Now, let us move on to the third representation 2's complement which in fact is an extension of 1's complement and in fact, this is the most widely used representation today. So, here again the positive numbers are represented as in sign magnitude form, for negative numbers you represent them in something called 2's compliment form.

Now, what is 2's complement form, the definition is simple well earlier we had seen what is meant by 1's complement, just flip all the bits you do 1's complement, then you add 1 to the number. This is defined as 2's complement right. Complement every bit of the number and then add 1 to the resulting number. So, here again the most significant bit will indicate the sign 0 for positive, 1 for negative.

$$2' \, complement \, of \, a \, binary \, number, X = 1' s \, complement \, of \, X + 1$$

(Refer Slide Time: 23:24)



Example for n=4

Decimal	2's complement	Decimal	2's complement
+0	0000	-8	1000
+1	0001	-7	1001
+2	0010	-6	1010
+3	0011	-5	1011
+4	0100	-4	1100
+5	0101	-3	1101
+6	0110	-2	1110
+7	0111	-1	1111

To find the representation of, say, -4, first note that

+4 = 0100

-4 = 2's complement of 0100 = 1011 + 1 = 1100

0100 : : 1011 + 1 = 1100

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | Switching Circuits & Logic Design

Let us see how the representation works. So, again an example for n equal to 4 so, for the positive numbers it is same as sign magnitude or 1's complement form no difference. But for negative number there is a difference. Let us again take the example for minus 4, plus 4 is 0100.

So, when you want to represent minus 4 like here we first we have to take the 2's complement of plus 4, plus 4 is this. 0100 was plus 4, you first take the ones complement 1's complement means you flip the bits 1011; 1011 means what in decimal 8 plus 2 plus 1 13, 8 plus 2 plus 1 11. You add 1 to it. So, we shall be taking about rules of addition later, if you add 1 to it will become 12, 12 is 1100. So, 1100 will be the representation of minus 4.

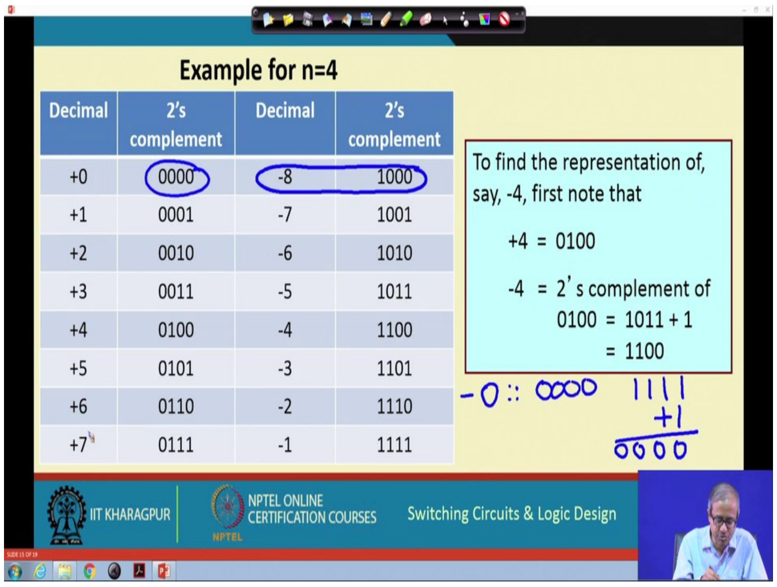
$$Signed\,representation\,of\,+4=0100$$

$$1'\,scomplement\,of\,+4=1011$$

$$2'\,scomplement\,of\,+4=1011+1=1100=-4$$

Now, another advantage here is that, well if you just look at the issue of 0, earlier we had the issue of 0, that 2 representations ok.

(Refer Slide Time: 24:55)



Example for n=4

Decimal	2's complement	Decimal	2's complement
+0	0000	-8	1000
+1	0001	-7	1001
+2	0010	-6	1010
+3	0011	-5	1011
+4	0100	-4	1100
+5	0101	-3	1101
+6	0110	-2	1110
+7	0111	-1	1111

To find the representation of, say, -4, first note that

+4 = 0100

-4 = 2's complement of 0100 = 1011 + 1 = 1100

-0 :: 0000 $\begin{array}{r}1111\\+1\\\hline 0000\end{array}$

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Switching Circuits & Logic Design

Let us try to compute minus 0, what is minus 0, you take plus 0, you take the 1's complement add 1 to it. If you want to add 1 to it you will see that in 4 bits the result will again become all 0's, which means minus 0 will also be having the same representation.

$$Signed\ representation\ of +0=0000$$

$$1^{'}s complement\ of +0=1111$$

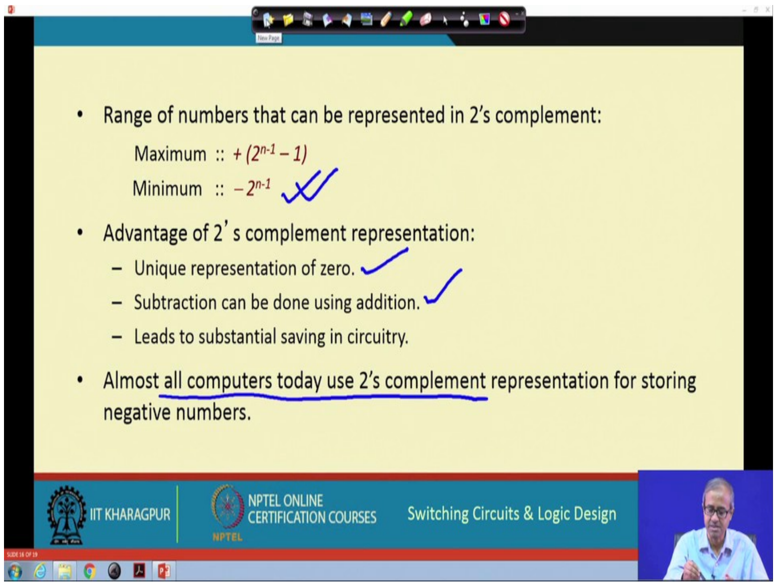
$$2^{'}s complement\ of +4=1111+1=0000=-0$$

So, there will be no 2 separate representations of 0, but what will this extra combination indicate, here I can add one additional number minus 8, this combination 1000 will indicate minus 8. This we shall see later also how this is coming, but in 2's complement this is one advantage. On the negative side, I can represent one extra number, on the positive side 0 to 7, in the negative side minus 1 to minus 8.

$$Maximum\ number=+\left(2^{n-1}-1\right)$$

$$Minimum\ number=-2^{n-1}$$

(Refer Slide Time: 26:09)



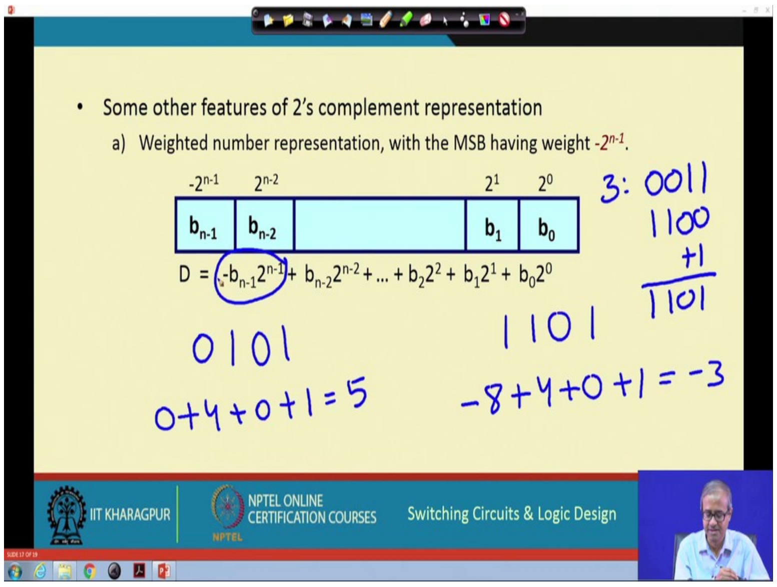
- Range of numbers that can be represented in 2's complement:
  - Maximum :: $+(2^{n-1}-1)$
  - Minimum :: -2^{n-1}
- Advantage of 2's complement representation:
  - Unique representation of zero.
  - Subtraction can be done using addition.
  - Leads to substantial saving in circuitry.
- Almost all computers today use 2's complement representation for storing negative numbers.

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Switching Circuits & Logic Design

So, the first difference is that the range of number is increasing by 1 on the minimum site we can represent 1 extra number. And this is because we have a unique representation of 0 and, just like ones complement, we can here also we can do subtraction by addition and, as I had said almost all computers today use 2's complement representation.

So, when we talk about representing negative numbers today. So, invariably we shall be talking about 2's complement representation ok.

(Refer Slide Time: 27:00)



- Some other features of 2's complement representation
  a) Weighted number representation, with the MSB having weight -2^{n-1}.

-2^{n-1}	2^{n-2}		2^1	2^0
b_{n-1}	b_{n-2}		b_1	b_0

$$D = -b_{n-1}2^{n-1} + b_{n-2}2^{n-2} + \ldots + b_2 2^2 + b_1 2^1 + b_0 2^0$$

$3: 0011$

1100

$+1$

1101

0101

$0+4+0+1=5$

1101

$-8+4+0+1=-3$

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Switching Circuits & Logic Design

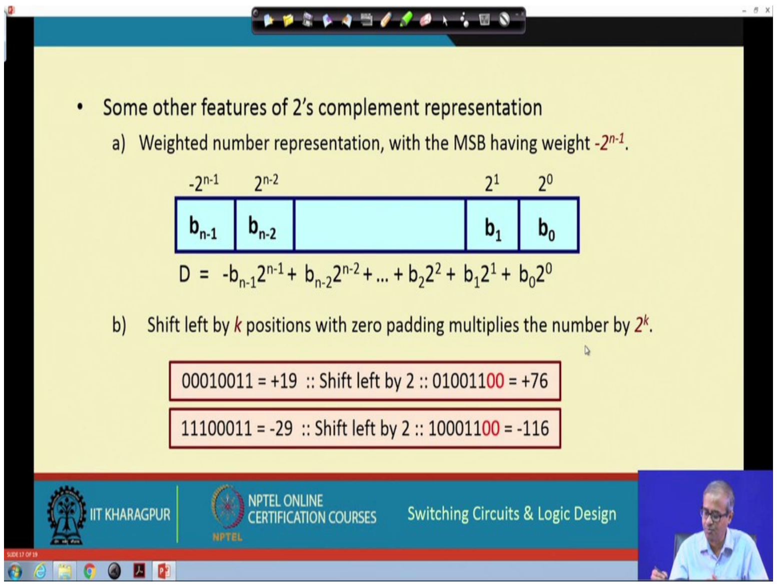
Let us talk about a few other features of 2’s complement representation, well this may not be very obvious, but when you are calculating the value of a 2’s complement number this can help. Suppose I have a n bit number I have a n bit number representation, in 2’s complement. Let us let us say in 4 bit I take two examples, let us say 0101 and let us say 1101. I want to find out its value. So, a simple rule is each bit position will be having weights, you add them up most significant bit will also be having a weight, but negative.

So, when I talk about 0101, 0 multiplied by anything is 0. So, the value will be 0 plus 4 plus this again 0 plus 1 which is 5. 2 to the power 0 $\overset{2}{\underset{¿}{(¿¿0)}}$, 2 to the power 1 $\overset{2}{\underset{¿}{(¿¿1)}}$, 2 to power 2 $\overset{2}{\underset{¿}{(¿¿2)}}$, 2 to the power 3 $\overset{2}{\underset{¿}{(¿¿3)}}$. But here because the most significant bit is 1 it will be minus 8 plus 4, 0 is 0 plus 1 so, this comes to minus 3. So, you can check minus 3, you take plus 3, plus 3 is 0011, take the 1’s complement, add 1 it becomes 1101, the same 1101. So, this is a very quick way of finding out the value of a 2’s complement number, you again follow that weighted some principle, only for MSB the weight is negative this is the only difference ok.

$$0101=0+4+0+1=5$$

$$1101=-8+4+0+1=-3$$

(Refer Slide Time: 28:54)



- Some other features of 2's complement representation
  a) Weighted number representation, with the MSB having weight -2^{n-1}.

$$\begin{array}{|c|c|c|c|c|} -2^{n-1} & 2^{n-2} & & 2^1 & 2^0 \\ \hline b_{n-1} & b_{n-2} & & b_1 & b_0 \\ \hline \end{array}$$

$$D = -b_{n-1}2^{n-1} + b_{n-2}2^{n-2} + \ldots + b_2 2^2 + b_1 2^1 + b_0 2^0$$

  b) Shift left by k positions with zero padding multiplies the number by 2^k.

00010011 = +19 :: Shift left by 2 :: 01001100 = +76

11100011 = -29 :: Shift left by 2 :: 10001100 = -116

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NPTEL
Switching Circuits & Logic Design

The other interesting thing is that when you have 2's complement representation. So, if you shift left a number by k positions, this is equivalent to multiply the number by 2 to the power k. Let us take some example, let us say in 8 bit representation I am showing plus 19, this is 1 2 4 8 and 16 so, 16 plus 2 plus 1 is 19. So, if I shift left the number by 2; that means, 10011 is getting shifted and 2 0's are inserted on the right. So, if you evaluate this number again 1, 2, 4, 8, 16, 32, 64, so 64 plus 8 plus 4 become 76. So, 19 multiplied by 4 is 76, 2 to the power 2.

$+19 = 00010011$

¿ $by\, 2\, bits = 01001100$

¿ $64+8+4$

¿ $+76$

¿ $(+19) \times 4$

So, even take the case of negative numbers, minus 29 is this you can check this, these minus 29 shift left by 2 shift left again 2 0's are inserted, it is minus 116 minus 29 multiplied by 4 is minus 116. So, shift left by every position is equivalent to multiplying a number by 2, if you shift left twice multiply by 2 square.

$-29 = 11100011$

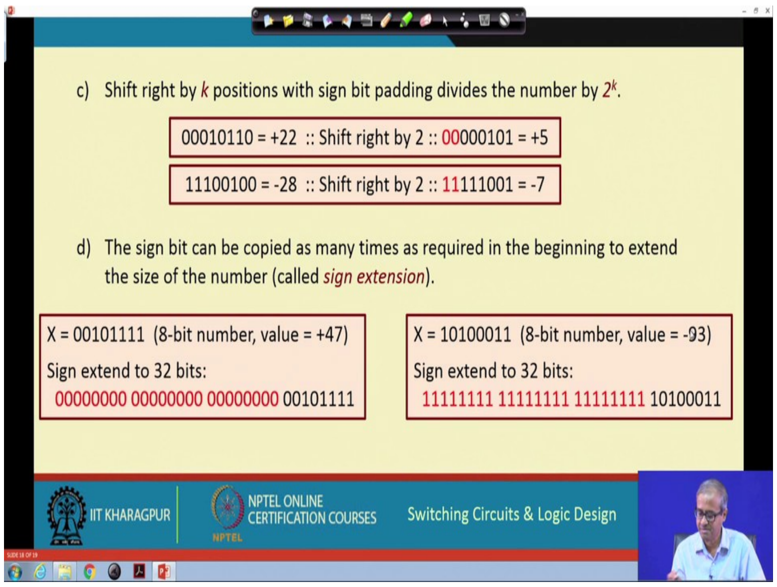
$¿ by\, 2\, bits = 1\,0001100$

$¿ -128 + 8 + 4$

$¿ -11\,6$

$¿ (-29) \times 4$

(Refer Slide Time: 30:20)



c) Shift right by *k* positions with sign bit padding divides the number by 2^k.

00010110 = +22 :: Shift right by 2 :: 00000101 = +5

11100100 = -28 :: Shift right by 2 :: 11111001 = -7

d) The sign bit can be copied as many times as required in the beginning to extend the size of the number (called *sign extension*).

X = 00101111 (8-bit number, value = +47)
Sign extend to 32 bits:
00000000 00000000 00000000 00101111

X = 10100011 (8-bit number, value = -93)
Sign extend to 32 bits:
11111111 11111111 11111111 10100011

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Switching Circuits & Logic Design

Similarly, if we shift right it means dividing by 2, if you shift right by k position it means dividing by 2 to the power k. So, again I give an example, there is a number this represents plus 22, this you can check. If you shift right by 2, 0's will be inserted on the left so this becomes 5 so, 22 divide by 4 is 5 point something, if you ignore the decimal part, it is 5. But only one thing if the number is negative and when you shift right, you shift the sign bit. Do not shift 0's, you pad it with this sign bit. Because, the number was negative, let the number remain negative.

$+22 = 00010110$

$¿ shift\, by\, 2\, bits = 00000101$

$¿ 4 + 1$

$¿ 5 (ignore\, the\, fractional\, part)$

$¿(22) \div 4$

So, it was minus 28 shift by 2, this number becomes minus 7, this you can check. So, again we are dividing by 4. So, shift right means division, shift left means multiplication by some power of 2 and one last trick let me tell you that just as this example shows when some sort of so, you replicate or copy the sign bit as many times as you want without changing the value of the number, this is called sign extension, like you consider an 8 bit number let say plus 47 in 8 bits, this is plus 47 you can verify 1, 2, 4, 8, 16, 32, 32 plus 8 plus 4 plus 2 plus 1. Now, suppose I want to represent this number in 32 bits. So, I have this representation and I add 24 0's in the beginning, this is called sign extension.

$-28 = 11100100$

$¿ \textit{shift by } 2 \textit{ bits} = 11111001 (\textit{pad with sign bit})$

$¿ -128+64+32+16+8+1$

$¿ -7$

$¿ (-28) \div 4$

So, you can check this number will also represent plus 47, the same thing holds for a negative number also. The only rule is that you are replicating the sign bit not adding 0 like here, the sign bit was 1, this is the representation of minus 93 this also you can check. So, you can replicate this sign bits, so many times, your number will remain still minus 93. So, when you change the size of a number you can freely replicate the sign bit and add it in the beginning as many times you want, this is a big advantage of 2's complement representation ok.

So, with this we come to the end of this lecture. So, we shall continue with our discussion in the next lecture again.

Thank you.