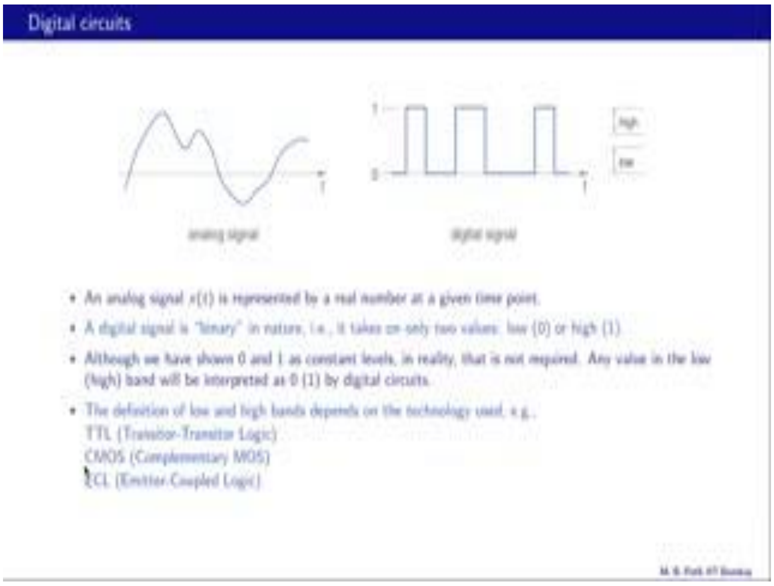
Basic Electronics

Prof. Mahesh Patil
Department of Electrical Engineering
Indian Institute of Technology, Bombay

Lecture – 55
Introduction to digital circuits

Welcome back to Basic Electronics. In this lecture, we will start with a new topic namely digital circuits. We will begin with an introduction to the digital domain and look at why it is advantageous. We will follow that up with the basic logical operations and then some identities related to logical operations and functions. So, let us begin.

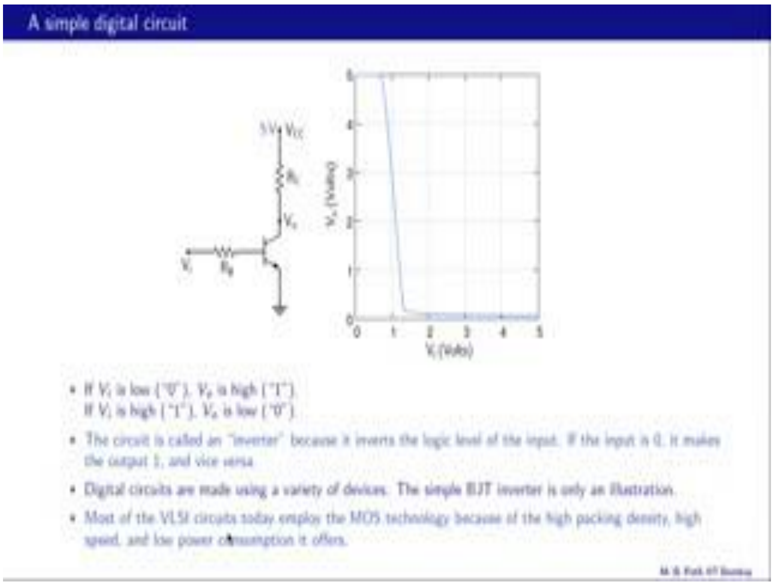
(Refer Slide Time: 00:44)



We will now discuss digital circuits. And to begin with let us talk about analog signals and digital signals. An analog signal x of t is represented by a real number at a given time point, for example, in this signal at this specific time point, the signal value is given by this number at this time the signal value is given by this number and so on. Whereas, a digital signal is binary in nature, what is the meaning of binary that means, it takes on only two values low or high a low value or a higher value. The low value is denoted by 0 and the high value is denoted by 1. Typically, this low value is 0 volts and the high value is 5 volts.

And we should remember that although we have shown 0 and 1 has constant levels these levels here; in reality that is not really required. Any value in this band here marked low will be interpreted by a digital circuit as a low value or a 0. And any value in this high band will be interpreted as a 1. And the definition of these low and high bands these depends on the technology used. And here are some examples transistor-transistor logic, which is based on the BJT technology, complementary MOS based on the MOS technology emitter coupled logic, this is also based on BJTs and there are a few others. And the most popular is the complementary MOS technology today and we will see why.

(Refer Slide Time: 02:52)

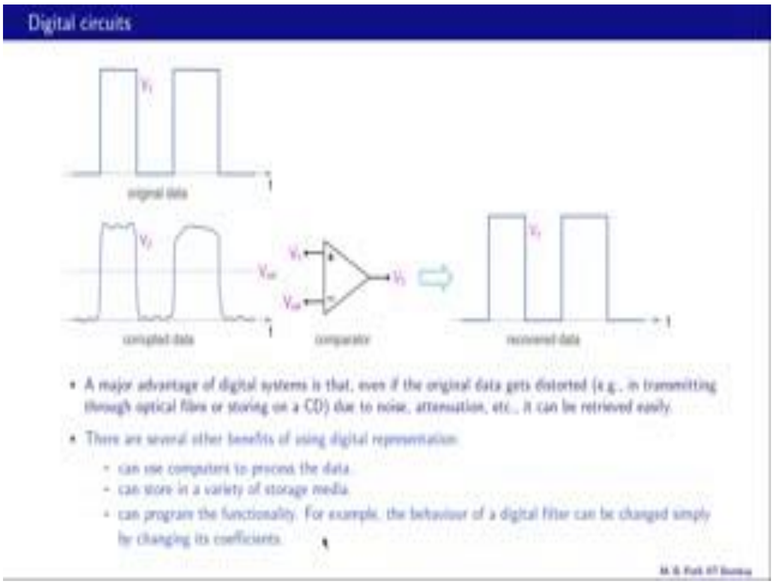


Let us take another look at this BJT circuit, which we have looked at earlier, but now we will look at it in a different context namely as a digital circuit. This is the input voltage that is the output voltage. And this graph here shows V o as a function of V i. If V i is low; that means, somewhere in this region then V o is higher as we can see here is 5 volts; and if V i is high somewhere in this region then V o is low something like 0.1 volts or 0.2 volts. Because of this functionality, this circuit is called an inverter; it inverts the logic level of the input. If the input is 0, it makes the output 1 and vice versa.

Now, we should remember that digital circuits are made using a variety of devices and technologies. And this simple BJT inverter is only an illustration; this circuit is not really used in practice, because it has several limitations. Most of the VLSI circuits today employ the MOS technology; in particular the complimentary MOS or CMOS technology, because of the high packing density high speed and low power consumption that it offers.

And you are probably familiar with the numbers here. What do we mean by high packing density in modern microprocessor, we have tens of millions of transistors in a single chip or even hundreds of millions of transistors, so that is how high the packing densities are today and that is made possible because of the CMOS technology. These modern chips also work at very high clock rates that mean, high speed and they consume very small amount of power. So, for a chip containing tens of millions of gates the power consumption may be 50 watt or may be 100 watts and that is on a per transistor or per gate basis that power is very, very small. So, that is why CMOS technology is extremely common and popular today.

(Refer Slide Time: 05:29)

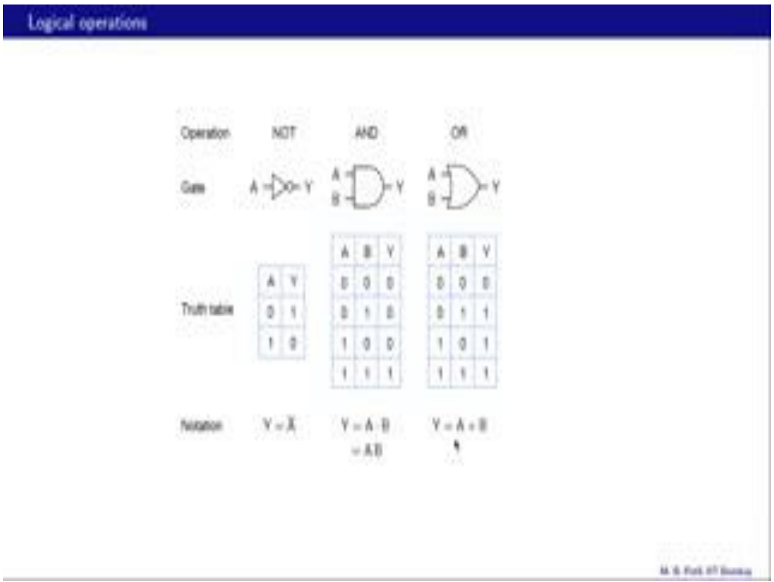


Let us discuss some of the advantages that digital circuits offer here is an example. Let us say we want to transmit this data 0 here, 1 here, 0 here, 1 here and so on. And as this data travels along let us say an optical fiber or through a satellite communication channel, it gets corrupted, these edges become less sharp, and also there is some noise or pickup riding on the original signal. And this is then the corrupted data. It turns out that we can easily recover our original data denoted by V 1 here from the distorted or corrupted data V 2 by passing V 2 through this comparator.

So, what do we have here the minus input of the comparator is connected to this level V ref where V ref is in between the low and high values here and the plus input is connected to V 2. When V 2 is higher than V ref, V 3 is high like that; and when V 2 is lower than V ref over here for example, then V 3 is low. So, in this manner, we recover our original digital data. So, this is a significant advantage of digital systems even if the original data gets distorted, it can be retrieved easily.

There are several other benefits of using digital representation or digital systems such as we can use computers to process the data that is a huge advantage, we use computers in many different ways, and the data being used is digital data. We can store the data in a variety of storage media such as CD or a pen drive or a hard disk or a magnetic tape and so on that is another benefit. We can program the functionality for example, the behavior of a digital filter can be changed simply by changing its coefficients; and these coefficients can be programmed, and this is possible only if we use the digital representation or digital systems.

(Refer Slide Time: 08:13)

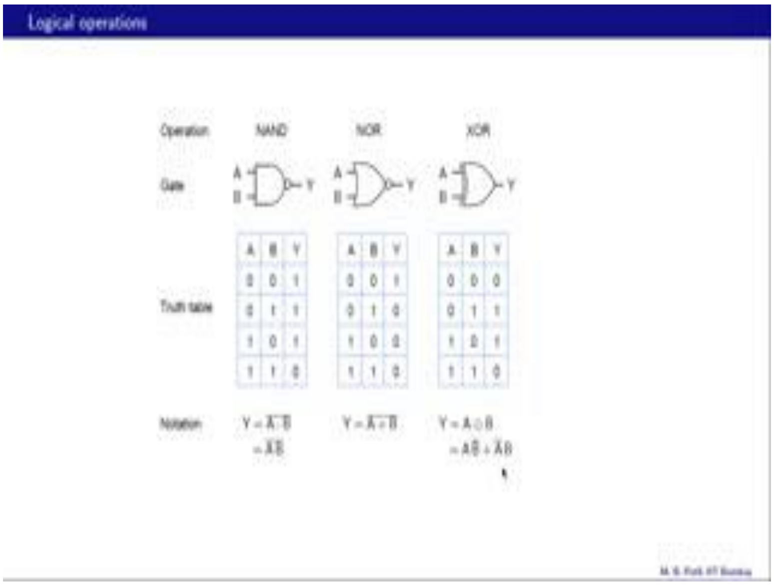


We have seen so far that digital or logical variable can take two values 0 and 1. And now what we want to do is to look at the operations that we can perform on these digital variables, those are called logical operations. And here are some basic operations NOT, AND, OR. Let us look at the NOT operation first. Here is the gate representation that is how the NOT gate is represented. A is the input to the gate which is a logical or digital variable and Y is the output also a logical variable. Here is the truth table for this gate, and what is the meaning of this term it means it is a table of values such that all input combinations are exhausted. In this case, there is only one input, so the only input combinations we have is A equal to 0, and A equal to 1. And the table essentially is a list of the output values for all of these values. So, if A is 0, then Y is 1; and if A is 1, then Y is 9. So, this is called the NOT operation and it is denoted by Y equal to A bar, a bar on top of this A here.

Here is the AND operation that is the symbol for the gate. Now the AND gate will have two inputs or more and here is the truth table for the AND operation. Now, since we have two inputs each input can be 0 or 1. So, therefore, we have 2 raised to 2 possibilities and therefore, there are 2 raised to 2 or four entries in this truth table and the output Y has to be specified for each input combination, so that is the meaning of this term here. So, how does the AND operation work. The result of the AND operation is 1, only if all the inputs are 1. So, here we have A equal to 1 and B equal to 1; and in that case Y is equal to 1; otherwise, it is 0, so 0 here, 0 here, 0 here. And the notation for the AND operation is Y equal to A dot B and very often that dot is not explicitly written, and it is understood.

Here is the OR operation that is the symbol for the gate; and the OR gate can also have two or more inputs. Here is the truth table. Again we have 2 raise to 2 or four combinations A equal to 0, B equal to 0, 0 1 1 0 and 1 1. Now the way the OR operation works is it is 0 the output is 0; if all inputs are 0. So, when A and B are both 0 then the output is 0; otherwise it is 1. So, if one or more inputs is 1, then the output is 1. And the way the OR operation is denoted it is with this plus operator, so that is our or symbol. So, Y equal to A plus B or A OR B.

(Refer Slide Time: 12:06)

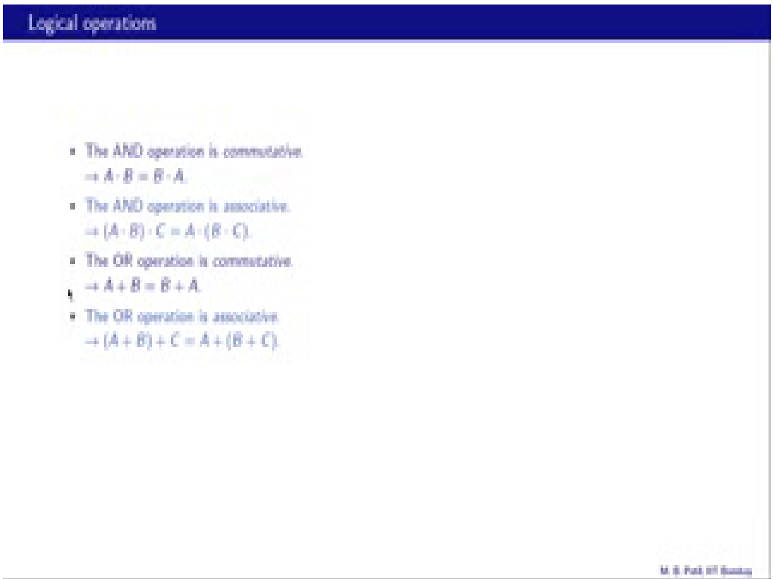


Here are a few other gates let us start with the NAND gate that is the symbol for the gate. And we notice that it has this and gate here and that is followed by this circle. Now, the circle represents the NOT operation and therefore, NAND is the same as AND followed by NOT and that is also reflected in the truth table here. You notice that it is exactly the opposite of the AND gate truth table. For the and gate we had 0 0 0 1, and now we have 1 1 1 0. This is the notation for the NAND gate a dot B and a bar over the whole thing. So, this is the AND operation and then that bar is the NOT operation, and very often we do not write the dot explicitly then it becomes A B and a bar over that.

Here is the NOR gate, and we see that it is a combination of OR which is the first part of the symbol followed by NOT which is this circle. So, the truth table for the NOR operation is exactly opposite to that of the OR operation and that is reflected in the table here for OR we had 0 1 1 1; and for the NOR operation we have 1 0 0 0. And this is how we represent the NOR operation Y equal to A OR B and then a bar on top to represent the NOT operation.

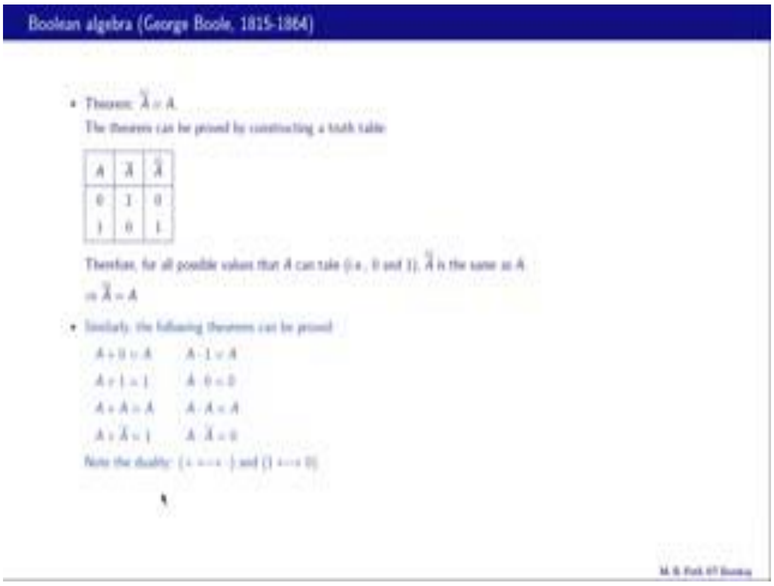
Next, let us look at the XOR gate here is the symbol; and this is the truth table. When A and B are both 0, Y is 0. When A and B are both one Y is 0 again, otherwise Y is 1. In other words Y is 1, only if A and B are different; and if a and B are identical, Y is 0, so that is how the XOR operation works. Here is the symbol for the XOR operation plus with a circle, and it is logically equivalent to this equation here A B bar plus A bar B note that this means A and B bar and this means A bar and B.

(Refer Slide Time: 14:56)



We will now look at a few properties of the logical operations such as AND, OR. The AND operation is commutative that means, it does not matter whether we evaluate A dot B or B dot A, they are the same. The AND operation is also associative that means, if we have A and B and that together ended with C, it is the same as B and C and that together ended with A. And the same thing also applies to the OR operation, OR operation is commutative; OR operation is also associative.

(Refer Slide Time: 15:39)

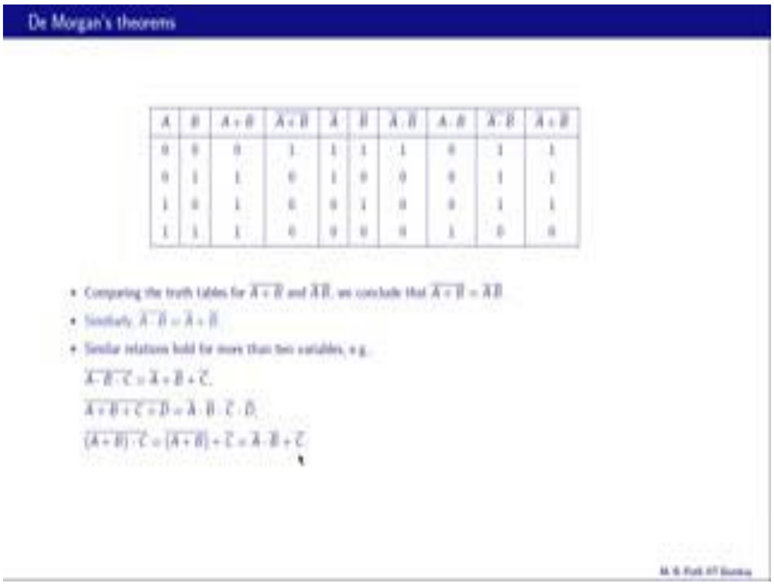


We will now look at a few theorems related to the logical operations we have seen. And this is a part of this larger field the Boolean algebra which was invented by George Boole in the 19th century. Here is a theorem which says A bar bar is equal to A, and it sounds a little trivial. Let us see how to prove that. The theorem can be proved by a constructing a truth table. So, we list all the possible values of the inputs, here there is only one input A, and therefore there are only two possibilities 0 and 1. So, A can be either 0 or 1. If A is 0, A bar is 1, and the bar of that which is this column here is 0. If A is 1, A bar is 0 and the bar of that is 1. Therefore, for all possible values that A can take namely 0 and 1, these two A bar bar is the same as A, because these two columns are identical and therefore, we can say that A bar bar is the same as A.

Here are some more theorems A OR 0 is A, A AND 1 is A, A OR 1 is 1, A AND 0 is 0, A OR A is A, A AND A is also A, A OR A bar is 1, A AND A bar is 0. This theorem will be useful to us when we evaluate logical expressions or minimize logical expressions and so on. So, it is a good idea to go over them again. Note that there is a duality between the OR and AND operation, and also between 1 and 0. Let us see what we mean by that.

Suppose, we know only this theorem and not this one in that case we can actually derive this theorem from here using this principle of duality. How does it work, we replace OR with AND that is plus with dot. So, we get A dot and then we replace 0 with 1. So, we have A dot 1 and that is A and that is this theorem here. Let us take some other example, this one. What does it say, it says A dot 0 is equal to 0, we replace this dot with plus and this 0 with 1, this 0 also with 1, then we get A plus 1 equal to 1 and that is this theorem. So, this duality is a very important principle and it can be useful.

(Refer Slide Time: 18:43)



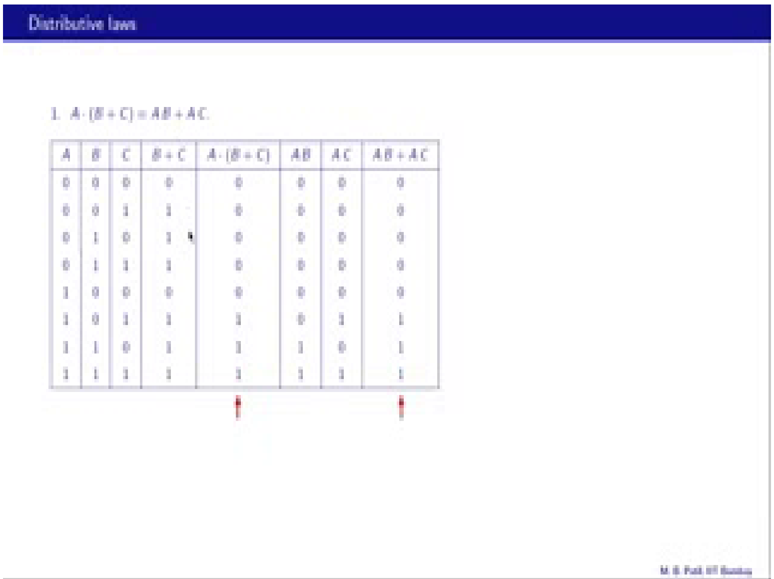
Let us now discuss De Morgan's theorems which are very useful in handling a logical expressions. And let us look at two variables A and B. Here we list all possible combinations of A and B, A equal to 0, B equal to 0, 0 1 1 0 and 1 1. And now we will construct each of these columns A OR B is 0 only if both A and B are 0 here, otherwise it is 1. A plus B bar just the inverse of this 1 0 0 0; A bar would be 1 1 0 0 like that. B bar would be 1 0 1 0 like that. A bar and B bar this will be 1 only when A bar is 1, and B bar is 1; otherwise it would be 0. So, it would be 1 here and 0 otherwise. A and B this would be 1, if A is 1, and B is 1; and 0 otherwise. What about A B bar just the inverse of this column; so 1 1 1 0. What about A bar plus B bar this would be 0, only if A bar is 0 and B bar is 0 and when does that happen that happens here a bar is 0 and B bar is 0. So, it would be 0 here and 1 otherwise.

Now, let us look at this columns and figure out the equivalence of some of these columns and that will give us De Morgan's theorems. For example, comparing the truth tables of A plus B whole bar which is here and A bar and B bar, which is here. We conclude that these two are the same this column is 1 0 0 0, this column is also 1 0 0 0, therefore, these two are equivalent. And similarly this expression A B whole bar is A bar plus B bar let us check that out this column is here 1 1 1 0, and this column is here 1 1 1 0, same. So, therefore, these two expressions are identical.

Similar relations hold for more than two variables also, for example, this theorem can be extended to this one for three variables. So, A B C whole bar is A bar plus B bar plus C bar. Similarly this theorem can be extended to more than two variables here we have four variables A plus B plus C plus D whole bar is A bar and B bar and C bar and D bar. And we can also apply these theorems two more complex expressions for example, we treat this as one variable, and then we can apply this theorem. So, instead of A, we have now A plus B here; and B it is C here when we apply this theorem we get A plus B whole bar plus C bar.

And now we can apply the De Morgan's theorem once again what is a plus B bar here A bar dot B bar, so that is A bar dot B bar plus C bar. So, this expression is equivalent to this expression here.

(Refer Slide Time: 22:38)



A	B	C	$B+C$	$A \cdot (B+C)$	AB	AC	$AB+AC$
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	1	1	0	1	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

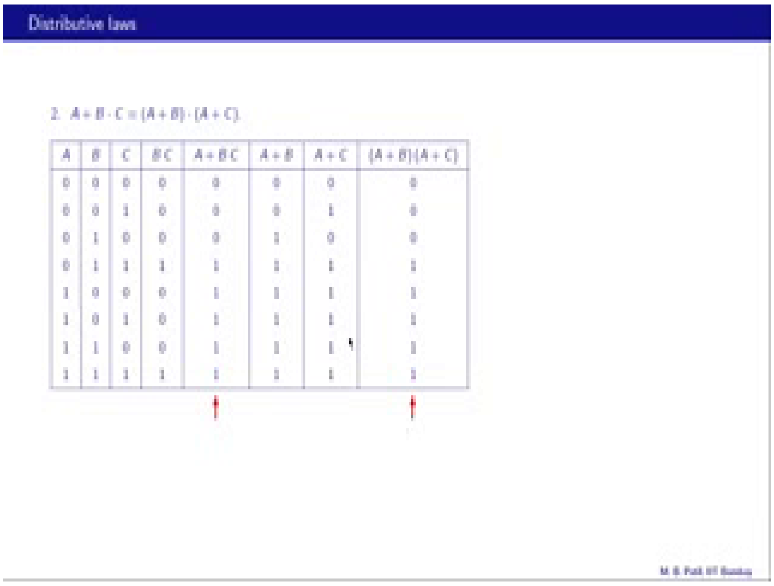
There are also distributive laws which hold in Boolean algebra. Here is one of them A and B or C is equal to A and B OR A and C, or we can read it as A B plus C equal to A B plus A C. And this particular law is very much like our other algebra the normal algebra that we use, but we need to prove this in the Boolean context. How do we do that, let us construct this truth table and we are going to check whether this column which is the left hand side here is the same as this column which is the right hand side.

So, let us get started. First we need to exhaust all the input combinations and in this case each input can take two values 0 and 1. And since there are three inputs, we have 2 raise to 3 possibilities that is 8 possibilities and it is good to do this systematically and a good way to do that is the following. Let C vary as 0 1 0 1 0 1 0 1, B has two 0s followed by two 1s then again two 0s and two 2s; and a four 0s and four 1s. If you do that then you will notice that we are exhausting all possible input combinations.

Let us now complete these columns one-by-one starting with B plus C. When is B plus C equal to 0 it is 0, if B is 0 and c is 0 and that happens over here and over here. So, in these two positions, we will have B plus C equal to 0; and in all other positions we will have B plus C equal to 1 as shown over here. What about this column a ended with B plus C, this is 1, if A is 1 and B plus C is also 1.

Let us look at a first A is 1 for these four positions; and among those positions B plus C is 1 for these three conditions, so that means we will have three ones here and all others would be 0s like that. What about this column A ended with B, this is 1, if A is 1 and B is also 1. Now, A is 1 for these four positions; and among those positions, B is 1 for these two positions. So, therefore, we will have 1 here, 1 here and all others 0s like that. What about A C, A needs to be 1, and C needs to be 1 and that happens here and here. So, we are those two entries as 1 and all others are 0. What about A B plus A C that is 1, if A B is 1 or A C is 1 or both are 1 and that happens as we can see in these three locations, so that is what we get in that column. And now if we compare, these two columns are identical and that proves this identity.

(Refer Slide Time: 26:15)



Here is the other distributive law that is A plus B C is equal to A plus B and A plus C. And this looks a little different than the distributive law that we are used to in our normal algebra. And in fact, this follows by the principle of duality from the first distributive law which we looked at in the last slide. And you should definitely check that out. Now we follow in the same procedure in this case and construct the left hand side that is here and the right hand side, which is here.

And then we compare these two columns. Let us check that out and we are not going to go through each column and how it is constructed and so on. We will just indicate the procedure that is B C A plus B C A plus B A plus C and finally, the right hand side. And if we compare these two columns, they are identical and that proves this identity.

In conclusion, we have seen the meaning of digital or logical variables and the operations associated with them. We have also looked at a few theorems or identities which are useful in manipulating logical functions that is all for now. See you next time.