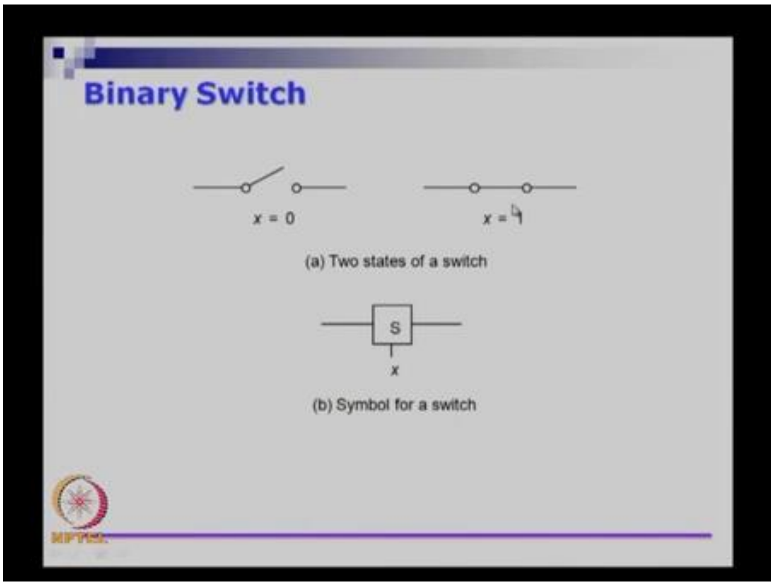
Digital Circuits and Systems

Prof. Shankar Balachandran
Department of Electrical Engineering
Indian Institute of Technology, Bombay
And
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Module – 2

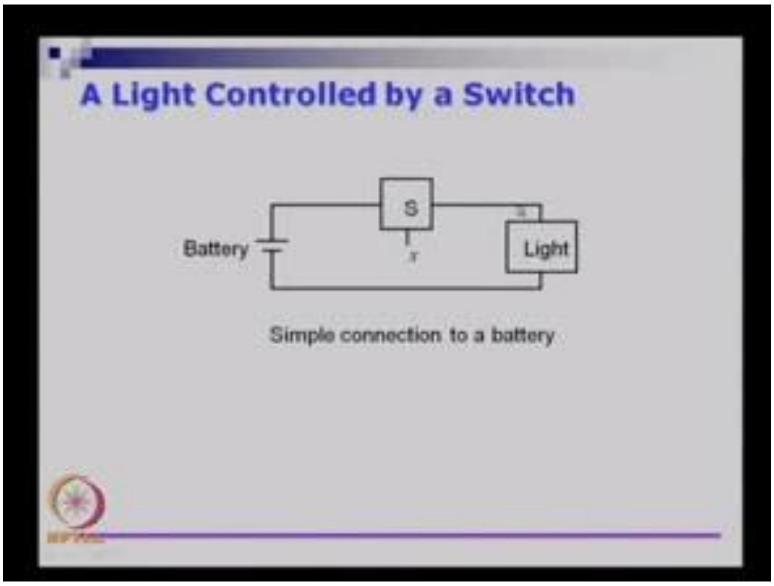
Welcome back. So, in this module, we will start looking at some of the building blocks – the basic building blocks for digital circuits. So, to understand what a digital circuit does, it is good to understand the notion of a switch.

(Refer Slide Time: 00:30)



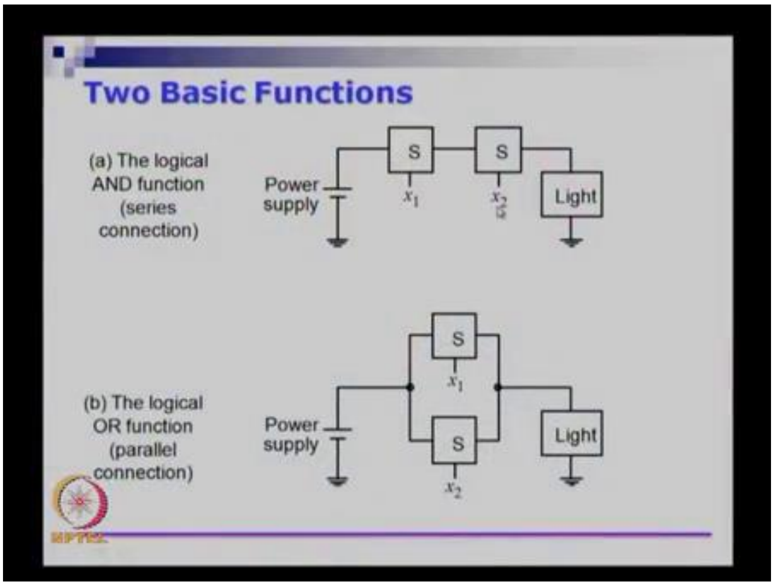
So, let us look at this picture here; this switch here and it is open in this position and it is closed in this position. We will start associating some symbols with this. If we call this switch as x; the x can be 0; which means the switch is open. So, there is no current it can pass through this. Or when the switch is closed, we will say that, this variable x, which is associated with this switch, takes the value 1. So, by taking a variable x for the name of the switch and the switch has two states; it is either 0 or it is 1. And we will start using this symbol from now on whenever we need it. So, we will put a symbol S inside and a control kind of thing called x. So, when x is 0, the circuit is open; and when x is 1, we will say that, the switch is in the close position. So, we will use these symbols for a while. This is a binary switch; which means it can take only two positions; it can be either like this or it can be like this.

(Refer Slide Time: 01:38)



So, now, let us take a simple electrical circuit. So, we want to see how to control this electrical circuit using a switch. For now, assume that, there is a battery that is connected to the switch, which is connected to the light and x is the switch's control. So, when x is 0, the switch is supposed to be kept open; and there will be no current from here through the light back to this. So, the light will not glow. So, when x equal to 0, the light will not glow. However, when we do x equals 1, this will close this circuit and there will be current passing through this light coming back here; which means the light will glow. So, when x is 0, this light will not glow; and when x is 1, the light will glow. It is as simple as that. And this is something that we do every day; we turn off lights and fans and whatever.

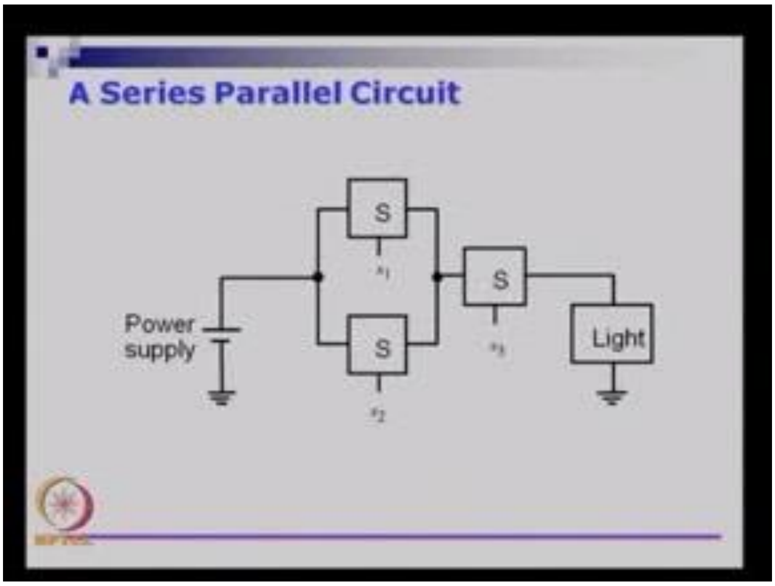
(Refer Slide Time: 02:31)



So, now, we will take some basic connections of these switches and see what happens. So, now, I am going to assume that, there are two switches that are in series. Let us call these switches x 1 and x 2. And x 1 can be either on or off; and x 2 can be again either on or off. And these two are connected in series. If this happens, then the only way that the light can glow is if x 1 should be on and x 2 should be on. So, if x 1 is on; then this circuit… So, from here to here this wire is connected. And when x 2 is on from here to here; it is also connected. So, only when both these things are on, the light will glow. So, which means x 1 and x 2 should both be on for the light to glow. So, whenever you have switches in series, we call them the AND function. So, AND is capitalized here; it is an operator also. So, x 1 and x 2 should be on for the light to be on.

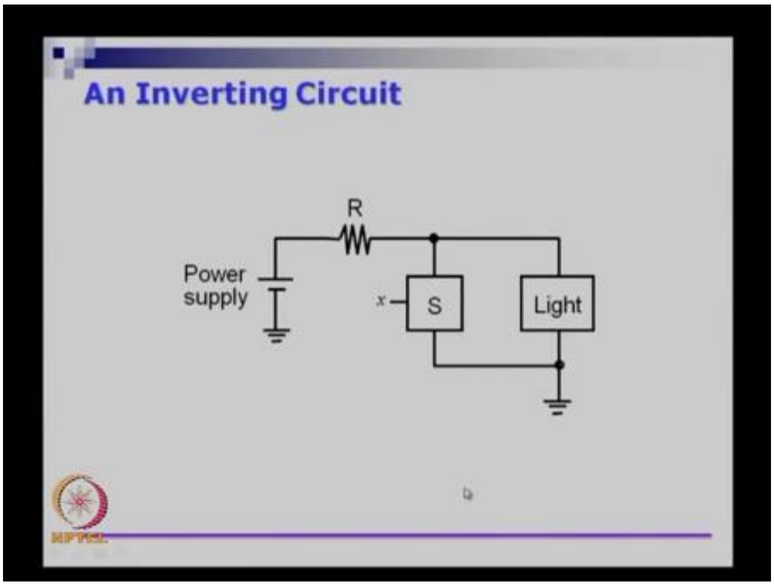
Look at the picture here. Again there are two switches. But, they are connected in parallel. So, x 1 is connected… This switch is connected in parallel to x 2. So, if x 1 is off and x 2 is also off; then there is no path from power to ground. However, if x 1 is on, there will be a current through switch x 1 to the light. If just x 2 is on; then there will be a current through this. If both x 1 and x 2 are on, there will be current through both of them and light will glow. So, what we want is in this setup, what happens is either x 1 or x 2 should be on for the light to turn on. This we will call the logical OR function. So, from this basic idea that this switch can be either on or off, we are now combining more than one switch to get some functions. So, we put two switches in series that gave us what is called the logical AND function. And when we put two switches in parallel, we will get what is called the logical OR function. So, this logical OR means either this or this should be on. So, at least one of them should be on for the light to be on. If all of them are off, then the light will be off. So, this is fairly easy.

(Refer Slide Time: 04:57)



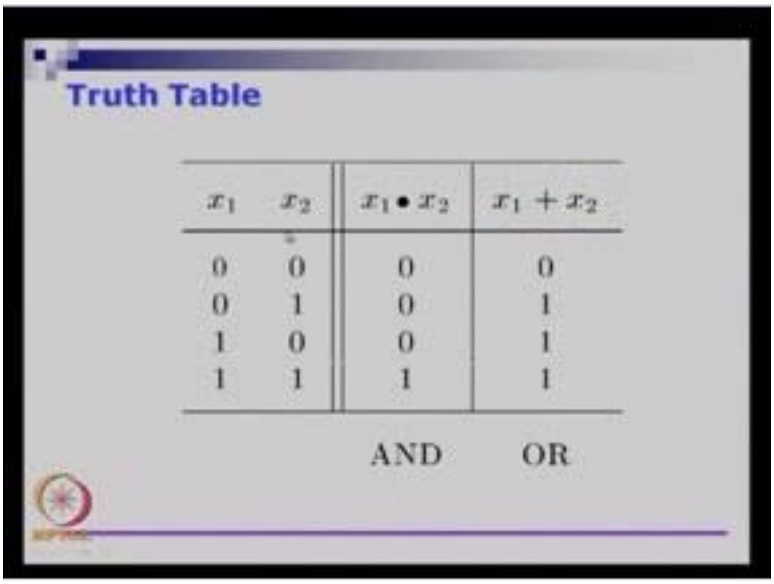
Now, let us take a look at this circuit. Just take a while and think about what are the conditions under which this light will turn on. So, there are three switches: x 1, x 2 and x 3. And this is the kind of setup. So, just take a while and think about what will make this light glow. So, for this light to glow, x 3 must definitely be turned on, because if x 3 is off; then whatever the state of these two switches are and if x 3 is turned off; then the light will not glow. So, we want x 3 to be on. And these two are in parallel; which means at least one of them must be on for the light to be on. So, the overall condition that we have is either x 1 or x 2 must be on; and x 3 must definitely be on. This is how you would say it in English. So, x 3 must definitely be on. And one of these should at least be on – either x 1 or x 2 or maybe even both if they are… That is okay, but at least we should have this condition that, these two should not be off at the same time. So, that is the way, which you can do what is called a series parallel circuits. So, we have two switches, which are in parallel. And these are in series with another switch here and that is controlled in the light.

(Refer Slide Time: 06:23)



Let us take a slightly twisted example. In this circuit, what we have is called an inverter. So, you have the power supply and let us say you have a resistor here. If the switch is off, then the current through the resistor cannot flow through this; it has to flow through the light and the light will glow. So, if the switch is off, the light will turn on. And if the switch is on; however, let us assume that, this is the path of least resistance and this is much higher resistance than this switch. If that is the case, then the current through this resistor will all flow through this switch and go here; the light will remain off. So, if x is on, light is off; if x is off, light is on; and this is called an inverting circuit; it reverses the sense of on and off. So, we have three different circuits that we have seen. We saw switches in series, switches in parallel and switches which give you the connection of these switches, which give you the inverting sense.

(Refer Slide Time: 07:39)

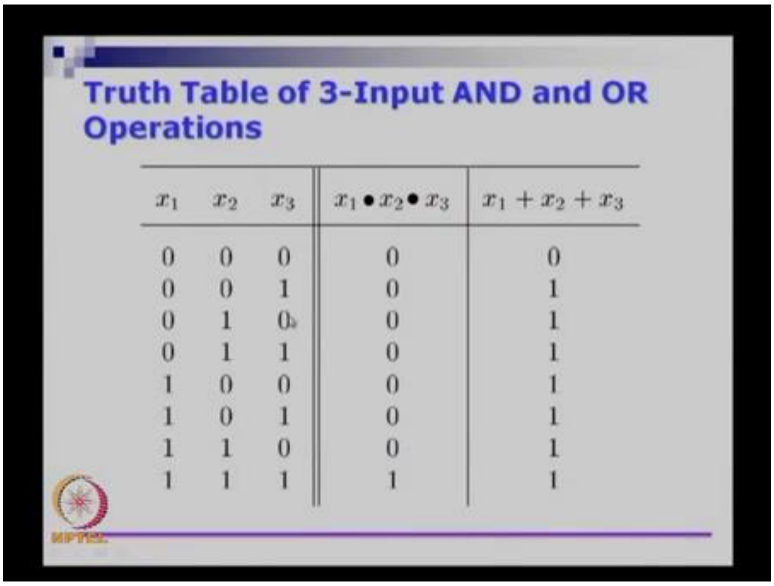


Now, let us see how we can build circuits based on these. So, the first thing we need to do is like in mathematics, sometimes we deal with symbols; sometimes we deal with pictures; sometimes we deal with tables and so on. Truth table is one essential way in which digital circuits are thought about. So, what we have is in this, on the left side, you see x 1 and x 2. Let us say these are two variables: x 1 and x 2. x 1 can take 0 or 1; x 2 can take 0 or 1. So, these are two switches let us say. And these two switches can be connected in series or parallel or whatever. What I am trying to show here is that, x 1 can be 0 and x 2 can be 0; x 1 can be 0 and x 2 can be 1; x 1 can be 1, x 2 can be 0; and x 1 can be 1; x 2 can be 1. So, these are the four different configurations in which two switches can be in. There are no other configurations that are possible for just two switches. For two switches, there are two squared or 2 power 2 combinations that they can take.

Now, let us look at what x 1 and x 2 gives. So, this column is for x 1 and x 2; and this column is for x 1 or x 2. So, if I want logical AND function; then when x 1 is on and when x 2 is on; x 1 and x 2 is on. If one of them is off, that happens in these three rows; if one of them is off, then the light will be turned off. So, these three will be off. That is what you get from series connection of the two switches. If we connected the two switches in parallel and again we use 0 for switch being off, and one for a switch being on; similarly, 0 for the lamp being off and 1 for the lamp being on. So, you have inputs here; and let us say this is the output. If we connected the two switches in parallel; if both of the switches are turned off, then the light will not glow; however, if at least one of them is turned on, the light will glow. That is what we get from the parallel connection of the switches. So, what we have here is an enumeration; or, all the possible choices of 0’s and 1’s that x 1 and x 2 can take.

And if you want a logical AND function on the inputs, then that is the set of values that we will get. If you want the logical OR function, this is the set of values that we will get. So, you can think of it as mathematical functions f of x 1 x 2. If it is x 1 and x 2; these are the set of values that it can take; and if it is x 1 or x 2; these are the set of values it can take. So, for AND, we will use the symbol dot x 1 dot x 2 or x 1 AND x 2. And for OR, we will use the symbol plus – x 1 plus x 2. So, do not think of this plus as addition of 1 and 1. So, do not ask me why this is not 2 and why this is only 1. So, think of it is an operator or a symbol that we have used to say that x 1 or x 2. So, always look at the plus; read it as OR also for the first few lectures or so. And then once you are comfortable with this notation that plus is OR and dot is AND; then you can say dot and plus. Till then I would recommend that you use AND and OR. So, as of now, this column is for x 1 or x 2; and we use the symbol plus for OR.

(Refer Slide Time: 11:22)



x_1	x_2	x_3	$x_1 \bullet x_2 \bullet x_3$	$x_1 + x_2 + x_3$
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	0	1
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

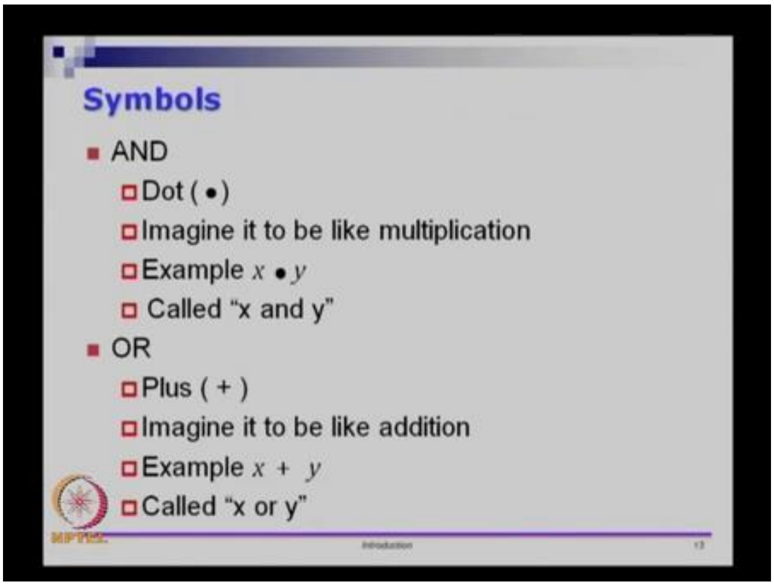
Let us look at this notion of truth table for three inputs. So, if I have three switches let say x 1, x 2 and x 3. The first thing you have to think about is what are the different values that you can have for x 1, x 2, x 3; what are the different combinations that you can have. So, x 1 can be 0 or 1; x 2 can again be 0 or 1; x 3 can be 0 or 1. So, these are all the possible combinations of x 1, x 2 and x 3. You can see that, all of them are off, all of them are on here; and several other combinations in between. So, what you see in this table is none of the input combinations are repeated. So, if you look at 0 0 0 as an input combination; that is not repeated anywhere else. Similarly, if I look at 1 0 1 as a combination, it is not anywhere else. So, each of the input combination is listed exactly once. So, in a truth table, what we are trying to see is we are trying to get the functions out of these inputs. So, when you see a double bar; that separates the inputs from the outputs. So, everything to the left of this double bar is input; everything to the right of that is output. So, in this case, there are three inputs: x 1, x 2 and x 3. All the possible combinations are given here; and the number of combinations with three inputs is 2 power 3 or 8 combinations. So, those are the set of values that you can take.

And here what we have is a three-input AND function and a three-input OR function. So, the interpretation of the input AND is – imagine three switches, which are in series. If all of them are turned on; only then the output will be 1; the lamp will glow. Even if one of them is off; which happens in these seven cases; the lamp will not glow. And therefore, the output is 0. So, that is the interpretation for a three-input AND. A three-input OR is three switches connected in parallel. So, even if one of them is ON; then the light will glow. The only case that, light will not glow is where all of them are off. So, if all of these are off; then the output is 0; otherwise, even if one of them is 1; then the output is 1. So, this is the interpretation for a three-input AND and a three-input OR.

Now, you can go and think about what happens if I have a four-input AND and a four-input OR and so on. If I have a four-input AND function, first of all, the number of inputs should be 4; let us assume that, x 1, x 2 and x 3 – another column added called x 4; then you will have another dot x 4 here. And the way it will be is you will have combinations starting from 0 0 0 0 all the way to 1 1 1 1; that is all this choice is. And everything, but the last row will contain a 1 for four-input AND. Similarly, for four-input OR, everything but the first one will contain a 1; only the first row should contain a 0 for a n input AND and OR function. So, you can think about or you can generalize this. So, for n-input AND, only the last row will be 1; for n-input OR, only the first row will be 0 provided you have arranged these inputs in a certain order. So, in this case, let us see the order in which these are written. So, think of this as a speedometer running from left side to right side. And this speedometer has only two digits: 0 and 1. So, this speedometer turns from 0 to 1. And now, it reads 0 0 1.

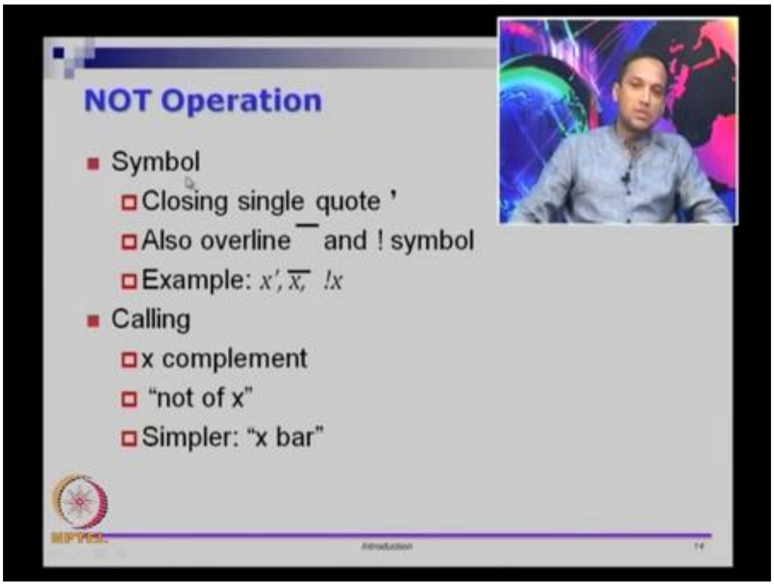
When the speedometer turns once more one step, this 1 turns over to 0; and the one to the left of it turns from 0 to 1. And again, when we go from here to here, it is 1 unit turn. So, the 0 turns to 1; this 1 remains as it is; and this 0 remains as it is. Then when this turns from 1 to 0; this also turns from 1 to 0; and this changes from 0 to 1 and so on. So, this is something that you must have seen in your scooters or bikes and so on. So, instead of 0 and 1, you would have 0 to 9 there. Whenever a digit roles from 9 to 0, the digit to the left of that will turn by one digit. That is what we see in bikes and cars and what not. The same thing happens here. Whenever something turns from 1 to 0; the one to the left of it will also turn one digit. And if you look at this example, this changes from 1 to 0; that is one turn. That will force this to turn from 1 to 0; which will enforce this one to change from 0 to 1. So, this is something that we see in your electric meter and speedometers and so on. There is some specific order in which things are written here; and the logical functions are described in terms of the corresponding input combinations. So, you have eight rows here and two columns. This is for function f 1, which is the logical AND of these three inputs; and f 2, which is the logical OR of these three inputs; and it has 8 rows and 5 columns.

(Refer Slide Time: 16:36)



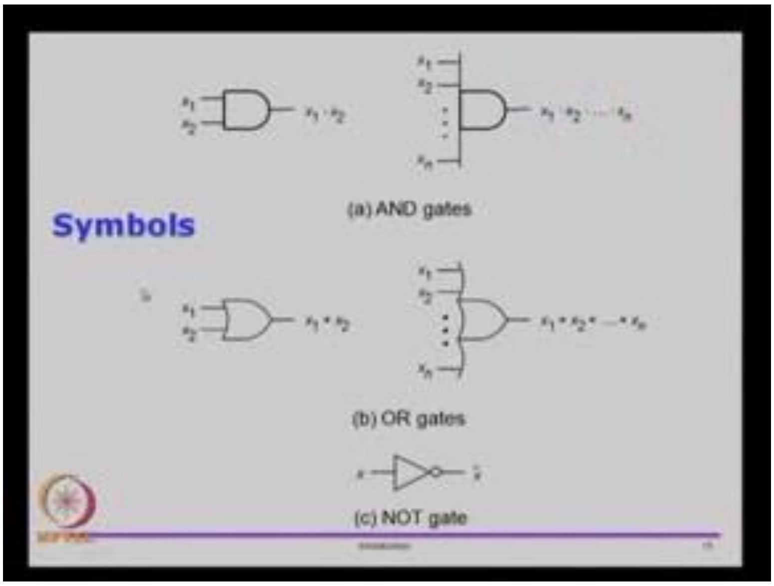
So, let us look at the symbols that we have used. For AND, we use the symbol dot. And you can think of it like multiplications. So, if you take numbers, sometimes you put a dot to imply multiplication. So, for example, if I put x dot y, you should read it as x and y. And for OR, we use the symbol plus; you can imagine this to be like addition, but it is not quite addition, because 1 plus 1 is actually 1 or 1. So, either it is 1 or another 1. The output is still 1; it is not 2. So, imagine it like to be like addition; it is not quite addition here. An example here is x plus y. So, x plus y is read as x OR y. So, till you are comfortable with this notations, I suggest that you read it as x AND x OR; instead of the dot and plus symbols.

(Refer Slide Time: 17:29)



Finally, we also need some symbol for the not operation; we cannot keep drawing things every time; sometimes we have to write it as text. So, we use the symbol either as closing quote or an over line or what is called a bang. So, examples are like these. This is x followed by a closing quote; x and a line above that or this bang x. We will call that either x complement not of x or usually we will call it x bar. So, x bar means whatever x takes, it is the complement of that value. So, complement of 0 is 1; and complement of 1 is 0.

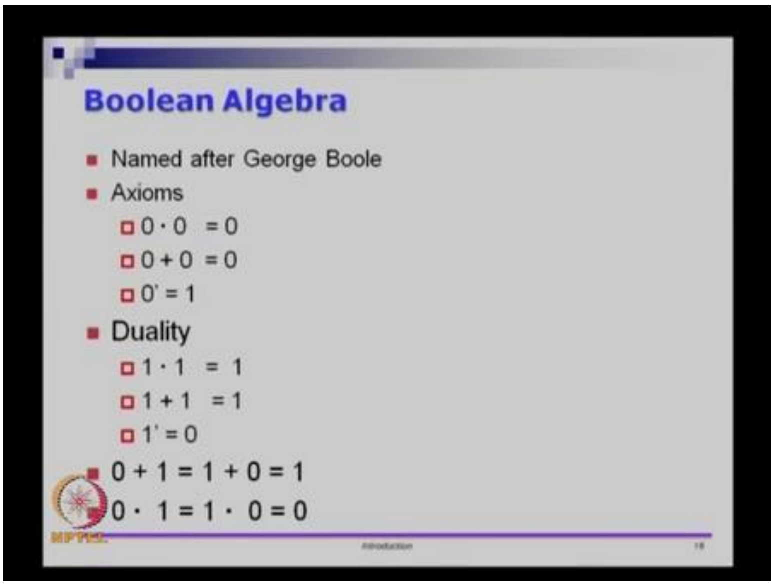
(Refer Slide Time: 18:08)



These are the standard symbols that are used for these basic operations. So, logical AND uses this symbol. So, it is a line and like semicircle kind of the thing here. In this case, what we have is a two-input AND. x 1 and x 2 are the two inputs that is given. And the output is x 1 and x 2. So, remember AND is dot. Similarly, this symbol is use for x 1 or x 2. So, x 1 or x 2. So, if one of the inputs is 1, the output will be 1; only when both the inputs are 0, the output will be 0 here. And for the NOT, if the input is 1, the output is 0; if the input is 0, the output is 1. So, we have x and a line above that. We either call it complement of x or x bar or NOT of x. So, what we see in these two pictures is what is called an n-input operation. So, instead of just two inputs, if I had n inputs that were coming in for... So, this is still the same symbol that we are using; the symbol that we have here; it is drawn as a line and a semicircle. So, what this means is all of the input should be 1 for the output to be 1. That is the meaning of logical AND; all of the input should be 1 for the output to be 1.

The logical OR; this is an n-input OR; what we have is at least one of the inputs should be 1 for the output to be 1. So, we can write it as x 1 plus x 2 plus so on up to x n. So, if you see this expression; what that means is at least one of them should be 1 for this expression to take the value 1 or true. Here x 1 dot x 2 and so on read as x 1 AND, x 2 AND, so on up to x n, is an expression. This expression is true only if all of these x 1 to x n are 1. For not gate, or for not operation, there is only one input at a time. So, x – if it 0, then the output is 1; if the input is 1, the output is 0; you cannot make a multiple input function out of inversion. So, each input can be inverted or not. So, we call these things gates, because in electrical circuits, we have symbols for resistors, capacitors and so on. Similarly, for digital circuits, we need symbols to represent them and to be able to draw them and so on. So, these are the symbols that we are going to use; and we will call them gates. You can think of it as a gate that opens the flow of current; you can think of it that way. And AND gate will let the current flow only when both x 1 and x 2 are on. An OR gate will let the current flow only when at least one of the x 1 or x 2 gets turned on. And a NOT gate will let the current flow if the input is 0; it will not let the current flow if the input is 1. So, you can think of it like a gate, which opens it up for current to pass through. So, do not extend this analogy too far. This is a simple way to remember why the term gate came about. So, gate allows the current to pass through.

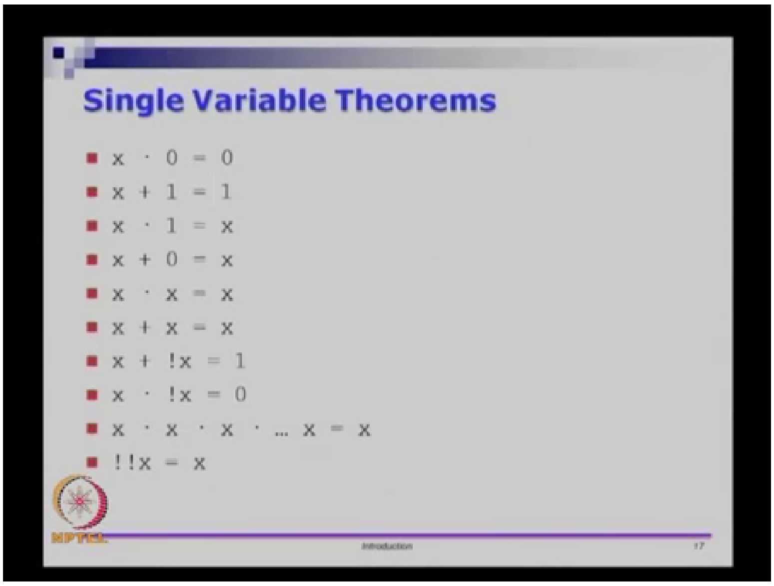
(Refer Slide Time: 21:18)



Let us look at what are called the loss of Boolean algebra. So, this is named after this person called George Boole, who lived in the nineteenth century. And the laws that he laid the foundations for are useful for digital circuits. So, we will start with the so-called axioms. Axioms are things that do not have to be proven; we take them for granted. The three basic axioms of Boolean algebra are as follows. This is just like what you saw in algebra and sixth grade. So, 0 and 0 is 0; 0 or 0 is 0; 0 complement is 1. These are three basic axioms of Boolean algebra. So, remember I read it as 0 AND, 0 OR, and 0 complement; I did not put dot and plus. The symbols are those; I do not use those words right now. So, these are three basic axioms.

We will assume that, we do not have to prove these things. These are the symbols and these are the meanings of these symbols defined. You can get what are called do else of these axioms by changing the 0's here to 1's and 1's here to 0's. You take these three axioms; wherever you see a 0, put a 1; wherever you see a 1, put a 0. If you do that, you get these three axioms. So, you look at this 1 AND 1 is 1; 1 OR 1 is 1; and 1 complement is 0. If you take these six axioms; it is already clear that, the way we looked at switches so far, this is quite consistent with that. So, if both the switches are off and if they are in series; then the light will not glow. If the switches are parallel and if both are off, then the light will not glow and so on. So, these are six basic axioms for Boolean algebra. And on top of that, we need these definitions. So, 0 or 1 takes the same value as 1 or 0; which is 1; 0 and 1 takes the same values as 1 and 0; which is 0. So, these are the basic things that we need in Boolean algebra. And what you see here is these are all constants; we actually have only 0's and 1's; we do not have any symbols like x's, y's and so on. These are the basic rules of Boolean algebra.

(Refer Slide Time: 23:42)

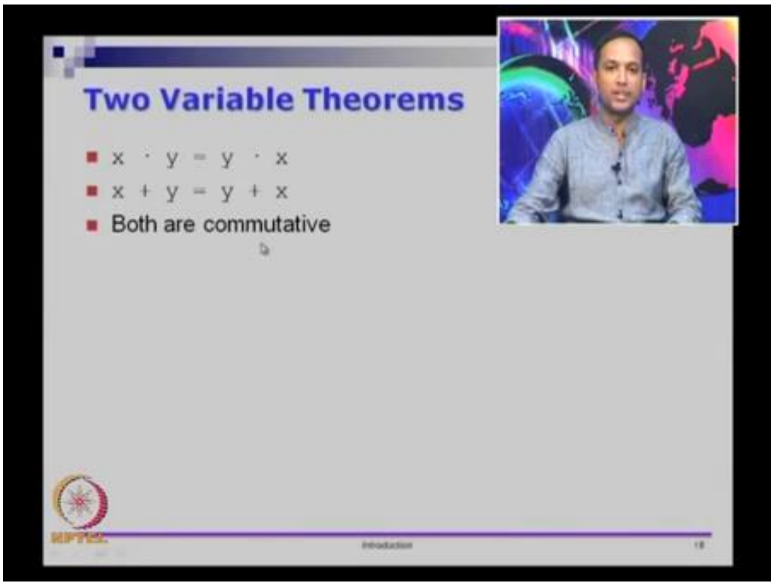


Now, it starts getting interesting only when we start putting some switches. So, let us look at these switches. Let us say I have a switch, which can either be on or off; we will call that switch as x – the position that it takes as x or the logical value that is attached to it as x. And these are what are called the single variable theorems. So, this is where it starts resembling what you did with basic algebra in sixth standard. So, you must have done things like five oranges and three apples cost 10 rupees; you would put 5x plus 3y equal to 10. So, those are called variable names. Here I am using the variable name called x; I may not know the value of x; but, these are all theorems that are true no matter what x takes. So, x can take either 0 or 1. So, in this case, x AND 0 is 0. What it says is if there are two inputs: 2 and AND operation; and if one of the inputs is 0; then the output is 0 no matter what x is. That is what you see here as a theorem.

Similarly, the way I would read this is if one of the input is a 1; it does not matter what the other input is, the output will always be a 1. So, this must be a 1. It is a mistake; I am sorry. This must have been a 1. Then x and 1. So, if do not know what x takes; x and 1 is still x. So, what it means is if I have an AND operation; whatever value this input takes is the value that the output will also take. Similarly, x plus 0 or x or 0 is x. x ANDed with x is x itself; x OR with x is also x itself and so on. So, one way in which you can prove all these theorems. So, remember in mathematics, you always prove theorems. So, to prove these theorems, you can substitute x equal to 0 or x equal to 1 and check whether the right-hand side is correct or not. So, for example, if I want to prove this one; I will put x equal to 0 first. So, 0 plus 0. We know that, from the axiom, 0 plus 0 is 0. And the right-hand side; x equal 0 is correct. If I put x equals 1; the right-hand side will be 1; left-hand side will be 1 or 0. That will also be 1 according to the axioms. So, by substituting values – all the values that x can take; you can show that, these theorems are all correct.

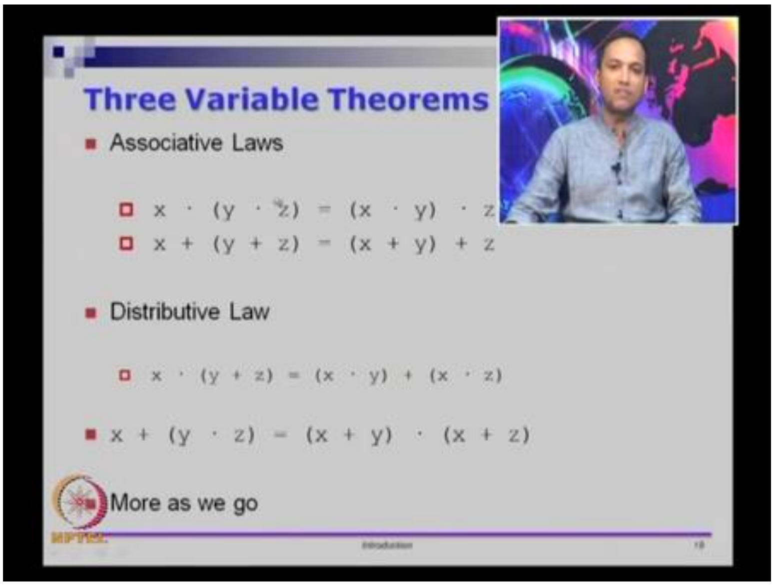
Some of the more interesting theorems are here. x OR complement of x is 1. What it says is if… If I read this in English; either x should be true or the complement of x should be true for the output to be true. That is what it is. So, if x is false, then NOT x is true. So, the output is 1. If x is true; then we have true or false. So, the output is 1 anyway. So, this is the slightly interesting theorem. This is also a slightly interesting theorem; you take x AND it with NOT of x; that is always 0. And this theorem says if I have x; if I have n of these x's. So, it is a same variable and AND it with itself many many times; the function is still the value that x takes. So, if I put 0 here; then I am looking at 0 and 0 and 0 and 0 and so on up to 0; we know that, that is 0. And similarly, if I put x equal to 1; here all of them are 1's. So, 1 AND 1 AND 1 and so on is still a 1. Finally, complement of complement of x is x itself. So, this is double negation if I invert an invert once more, then the output is the same as the input itself. So, these are single variable theorems.

(Refer Slide Time: 27:29)



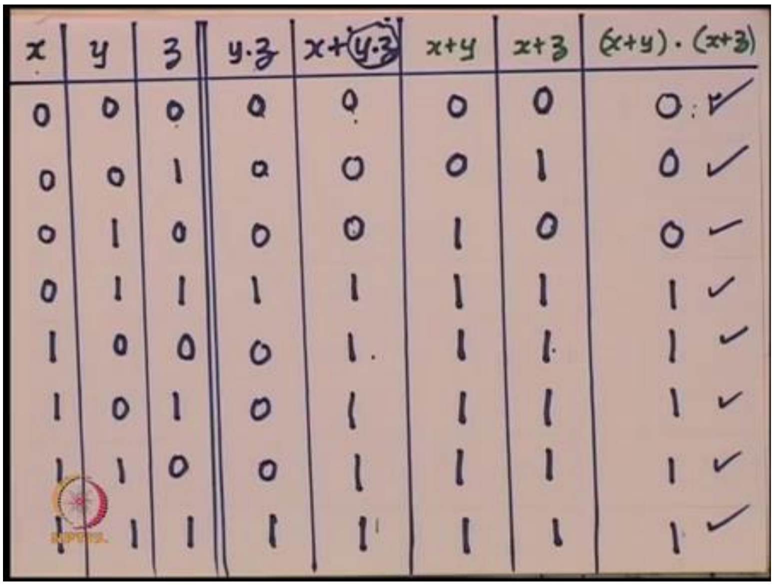
Then, we have two interesting two variable theorems. x and y should be the same as y and x. So, what that means is the order in which give the inputs does not matter; x and y is same as y and x. Similarly, x or y is a same as y or x. Again you can prove these things by riding a truth table down. I will show you an example really quickly. But, right now, just assume that, these are correct and these are both commutative; which means the order of the operations do not matter.

(Refer Slide Time: 28:04)



Finally, we have the so-called three variable theorems. Three variable theorems assume that, there are three switches: x, y and z. And they are connected using this fashion. If we put all of them in series; this is what it means. y is connected in series with z; and that setup is connected in series in x; the value that x and y and z takes as a same as the value that x and y will take and it with z. Similarly, x or y or z will take the same value as x or y or z. So, this is called associative law. Then there is distributive law; just like what we have for integers, this is distributive law; x and y or z is a same as x and y or x and z. So, you can see that, this dot is distributing over plus. And finally, we have this one slightly interesting theorem; which is true in the Boolean world. So, these things are also true; the integer world and… If you take real numbers and so on; however, this is true only in the Boolean world; x plus y and z is x plus y and x plus z. So, we will see many more of these things as we go; but, I want to conclude this lecture showing how to prove some of these theorems. So, I am going to take this as an example – x plus y and z. I want to prove that it is a same as x plus y and x plus z. So, right now, we do not have any other mechanism than truth table. We will see how we use the truth table to prove the correctness of this. So, I am going to start with this written up here.

(Refer Slide Time: 29:46)



x	y	z	y.z	x+(y.z)	x+y	x+z	(x+y).(x+z)
0	0	0	0	0	0	0	0 ✓
0	0	1	0	0	0	1	0 ✓
0	1	0	0	0	1	0	0 ✓
0	1	1	1	1	1	1	1 ✓
1	0	0	0	1	1	1	1 ✓
1	0	1	0	1	1	1	1 ✓
1	1	0	0	1	1	1	1 ✓
1	1	1	1	1	1	1	1 ✓

I am going to fill up the truth table in the sheet here. So, I have the variables: x, y and z. And I am going to put three inputs here. So, there are three inputs. And I am going to see this if you notice is the LHS and this one the last column here is the RHS. I am going to see if the columns match. So, let us see all the values that x, y and z can take. So, the first set of values is 0 0 0. Now, I am going to evaluate what y and z is. What is y and z? It is 0 and 0. We know that, 0 and 0 is 0; and x or y and z is taking the logical OR of this column with this column. So, 0 OR 0. So, that is a 0. So, this is the way you can fill up a truth table. So, let me fill up the rest of the table as follows. So, remember I am going to run like a speedometer 0 0 0 here; 0 0 1; then this speedometer turns from 1 to 0 here. Whenever this changes from 1 to 0; this should change from 0 to 1. So, this is the same. Then this changes from 0 to 1. This one remains the same and this one remains the same.

Now, when this changes from 1 to 0; this should also change from 1 to 0; but, because this is changing from 1 to 0; this should go to 1 and so on. So, if you continue this; the inputs will follow in this manner. These are the set of inputs. Let me show you how to prove this theorem that we started with. So, let us start with y and z first; it is taking this column and this column and ANDing them together. So, 0 AND 0 is 0; 0 and 1 is also 0; 1 and 0 is 0; 1 and 1 is 1; 0 and 0 is 0; 0 and 1 is 0; 1 and 0 is 0; 1 and 1 is 1. So, we have filled up these columns; but, that is not the complete LHS. We will see the LHS; LHS is taking the logical OR of this column and this column. So, 0 OR 0; that is 0; 0 OR 0; that is 0 again. 0 OR 0; that is 0 again.

Now, we have 0 OR 1. We know that, that is 1; then 1 OR 0; that is 0; that is a 1; 1 or 0;

that is a 1; 1 or 0 is a 1; 1 or 0 is a 1. So, I took this part first and then ORed it with x. So, now, we are done with the left-hand side. Let us look at how to get the right-hand side. Right-hand side has two terms; I am going to get these two terms separately and then AND them together. What is x OR y? I am going to look at the first two columns and write x or y. So, 0 OR 0 is 0; 0 OR 0 is 0; 0 OR 1 is 1; 0 OR 1 is 1; 1 OR 0 is 1; 1 OR 0 is 1; 1 or 1 is 1; 1 or 1 is 1. So, that gives me one of the terms. Let us look at the second term here; x plus z or x OR z. So, 0 or 0; that is a 0; 0 OR 1; that is a 1; 0 or 0 is 0; that is a 0; 0 or 1; that is a 1; 1 or 0 is a 1; 1 or 1 is a 1; 1 or 0 is a 1; 1 or 1 is a 1. Now, I will take these two terms; which are the terms that we have in the RHS. And I am going to put a logical AND operation in between. 0 and 0 is 0; 0 and 1 is 0; 1 and 0 is 0; the rest are all 1 and 1. So, we will have 1's everywhere. So, this path is the left-hand side; and this path is the right-hand side.

Now, let us go and check each of these rows. So, when input combination is 0 0 0 0; left-hand side is 0; right-hand side is 0. So, this is… So, this theorem is to be correct for 0 0 0; then 0 0 1. The left-hand side is 0 and the right-hand side is 0. So, this also correct. So, we look at the match of this column and this column; this column and this column. So, these are same. This and this are same; this and this are same and so on. So, all the five are 1 here; all the five are 1 here. So, all of these are correct. So, for all the possible choices of input that we have here; these two columns match; which means the left-hand side must be equal to the right-hand side. So, this is such a basic proof technique that you need to know. And if not for anything, you can always go back to what is called drawing the truth table to get the correctness. If I want to you to show that something is equal to something, you can always go to the LHS; get a truth table for it; go to the RHS; get a truth table for it and show that these two columns are equal. This will work for… If you have very few inputs, this will work well; this may not work well for bigger number of inputs or for other functions. But, at least there is a fallback option; this is a safe option that you can always resort to if you do not have any other technique to solve this problem. So, we will see more of this three-variable theorem and so on as we go. This brings me to the end of module 2 of this first lecture series for the week. I will meet you next time with more and more theorems and how it connects to digital circuits.

Thank you.