Digital Circuits and Systems

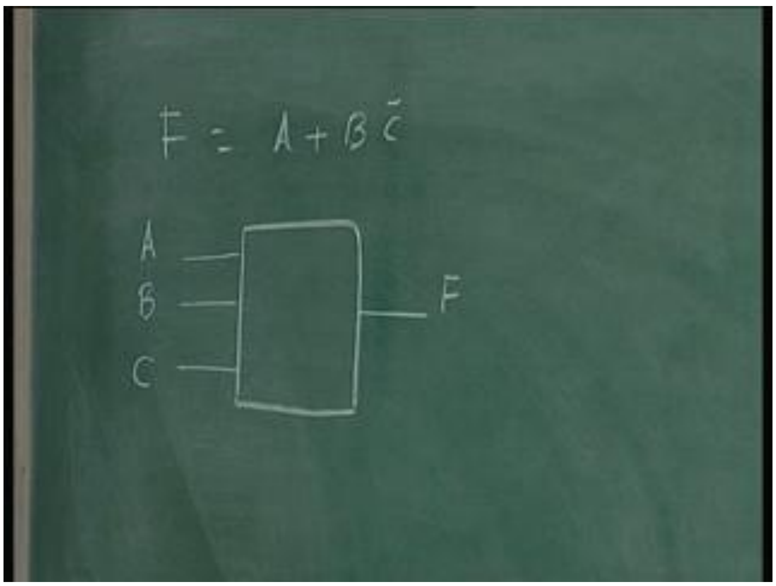
Prof. S. Srinivasan
Department of Electrical Engineering
Indian Institute of Technology Madras

Lecture - 5

Logic Simplification

In the last lecture we talked about logic functions and truth tables and how it can be implemented using gates. We took a simple example of a function F which is if I remember was A OR BAND NOT C.

(Refer Slide Time 3:31)



That means we said there is a digital system or a circuit we call it whatever there are three inputs A B C and an output F and this output F is true for this condition of the input combinations which is output is true if A is true OR if B is true and C is NOT true. And we were able to realize this using gates a NAND gate to get B and NOT C and an inverter to get NOT C out of C and an OR gate to combine A and this term. We just took the arbitrary gate structure and wrote the logic function and we also said that this is true under these conditions it could be a control for a particular operation any different situation and then we wrote the truth table with A B C as the input variables and the output F and we defined certain things like min terms, sum of products and so forth.

Many times either given a function before you implement it as a logic circuit with gates or given a logic circuit and you analyze the behavior we are not sure whether that is the simplest possible circuit in terms of number of gates and number of inputs of each of these gates to retain the same function.

If I enable to get a same function implemented using a smaller number of gates or with smaller number of inputs for a given gate I would prefer that because it is a saving in hardware. So that has to be some methods of finding out whether what we have is the simplest possible realization, if not what are the techniques we can use to reduce the given function or a given circuit to a simpler form.

As I said the other day there are two approaches; one is called the Boolean algebra the mathematical tool wherein you write the function and use some identities apply these identities to this function to the left hand side and right hand side and simplify it to the extent possible. But the problems with such an approach is we never know which is the minimum possible whether we reached a minimum because an identity may not strike you, somebody else may think of something which you may not be able to think of. You can never be sure of whether a particular minimization reduction is the simplest.

The other approach is the graphical method where there is a systematic technique applied to a given function till we reach a specific minimization beyond which we cannot go. But of course this graphical is sort of integrated into a mechanized or automatic procedure. So you automate the procedure so that you will not make a mistake. Hence it is not that graphical method it is called map method we will see that it is not that it is totally a different technique compared to a Boolean algebra.

Boolean algebra is the algebra based on the identity the logical behavior of these variables under different combinations and the same thing will be applied in the systematic way in a graphical sense using a map so that some sort of a tool to go systematically into the function and simplify it to the extent possible. And that is more practical, easier to implement and more than that once you want to do a program or automatic way of reduction using some computer programs such a technique which is step based technique an automatic procedure based technique becomes easier to implement in the program so that you can get the best possible results.

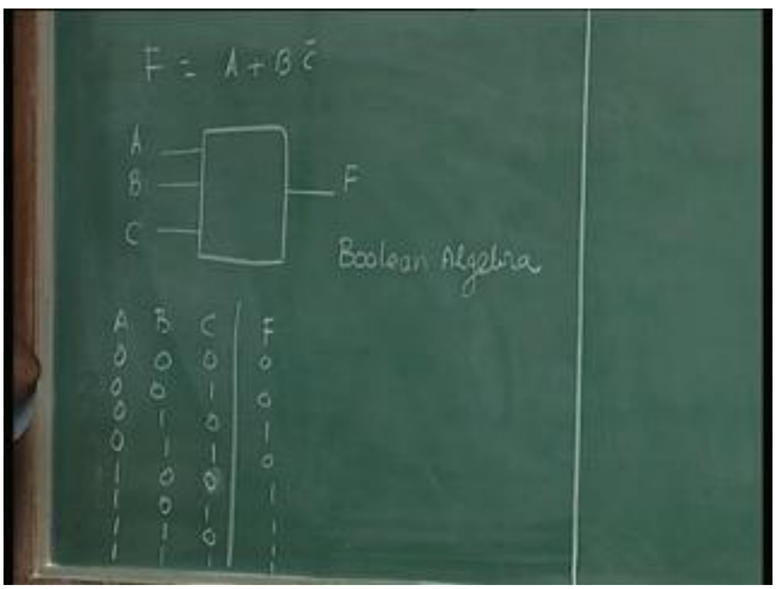
What we are seeing in the class are simple circuits with two inputs, three inputs, four inputs and a maximum of five inputs and not beyond that because we cannot handle a circuit like that in a classroom, in a given time, in a given space also becomes extremely complex then the principle and the concept which we are trying to teach you gets submerged in numbers, equations and variables that is why we always take simple examples. But then in practice in reality when you talk about digital systems as I said a digital system could be a micro processor, a computer or as I said a traffic controller or as I said aircraft navigation system or a missile launch system.

Think of those systems how many variables how many conditions will be there and how can you do the simplification if you do not a have systematic procedure to do that, that is why we go for the second method. But before going into the second method I want to introduce the first method because it is the basis on which the second method is built it is a mathematical method being a scientific basis on which reduction is based. So I will briefly introduce Boolean algebra and not use it very much for many reasons as I said the other day again.

Today the reduction at the gate level has become more or less not essential because you are at the days where we need to save gates because of the large technology availability of gates and the availability of circuits in which a lot of these functions can be put in a very simple small IC so there is a need for reduction at the gate level which not as great as it is used to be. Even then it is required because occasionally there are situations where you want to reduce the given function to a fewer number of gates for many reasons. That is one reason.

The second thing is we are going to use the other method which is going to be a more practical method extensively in the course for that reason also you don't have to learn Boolean algebra very much. But still as I said I want to introduce this concept. So let us again take the same example. this given system digital circuit which you want to implement you remember there are three inputs and the three inputs can have eight different combinations and for these combinations the output is true that means if A is true the output is true or if B is true and C NOT true B is true C NOT true but already this is true because of A is 1 so for the rest of the combinations A output is 0.

(Refer Slide Time 11:37)



This is the truth table we called it and each of this row is called a min term wherein all the inputs are present either in the true form or the complement form and you can write a function which is true form as a sum of products. That means (Refer Slide Time: 11:29) this is a sum of products. That means we have an output which is written as a series with terms which are AND functions. When you say product you give a mathematical analogy arithmetic analogy. Even though it is B and NOT C you call it a product B C NOT. When you read it you say B C bar so it is a product and even though it is an OR function we call it a sum function because of this plus sign. So this is a sum of product but you write it from the truth table we will get what is known as a canonical sum of product I said that. So we will first write the canonical sum of products and the canonical sum of products will have the min terms corresponding to each of these input combinations for which the output is true.

So when you write that you get the canonical sum of products so each of these product terms is called a min term. So F is true if A is NOT true B is true C is NOT true then on all possible combinations A is true B NOT true C NOT true plus A true B NOT true C true etc all the four combinations. So A true B NOT true C NOT true plus A true B NOT true C true plus A true B true C NOT true plus A true B true C true. These are the five combinations of inputs true or NOT true for which the output is true, the other combinations or the combinations for the output are false.

If I write in this form this is a canonical sum of products because you have all the possible combinations of the input combinations for which the output is true expressed in this form. And as I said this also called Y is called sum of products because it looks like a product term combined by a sum series. Even though it is NAND, NOR operations we are referring to AND is loosely referred to as a product operation and OR is loosely referred to as a sum operation and each of these terms is called a min term. There are eight min terms in this out of which for five min terms the output is true for three other min terms the output is false. So we write the sum function as sum of all the min terms for which the output is defined as true.

Now let me show you how to apply the Boolean algebra to this to get this form. So now let us use Boolean algebra. What is a Boolean algebra? After all it is a very simple thing. Again fortunately for us it is not a very complex mathematics because each variable can only take two values A and A bar so it is a binary operation. so in the binary operations the functions can be easily understood even though there may be a rigorous mathematical proof for some of these identities and theorems as we call it in Boolean algebra from understanding point of view it is only simply a two level you can always verify it by simply giving two different values for each of these things and verifying whether it is true or false.

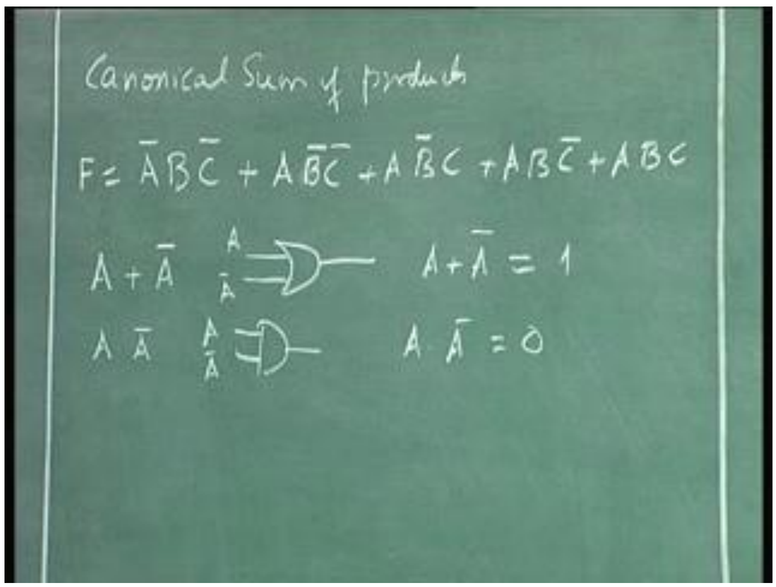
If you make a statement then you can quickly say whether it is true or false by giving values for all the functions all the variables in that function and each of these variables can only have two values so give all the possible values and verify it for all possible input combinations then the output is true. So I can verify this without any formula on mathematics by simply giving alternatively 0 one 0 one for A B C and then for each one of them whether it is true or false and then it is verified.

So what are the simple rules we are going to use? I am not going to introduce Boolean algebra in rigorous sense as I said this is not required. Just to give you an idea, flavor of this so that you should not think these are all some heuristic procedures which may or may not work under different conditions I don’t want you go with that impression that is why I am just introducing this to you as a concept as I said even though it is not important.

Suppose I have a function A and it has two values for all combinations of A OR A bar that means I have A in other words I have an OR gate in which two inputs are connected one as the function A the other is complement of that function the output has to be 1 because if one of them is true the output is true if A is true A bar is false and if A is false A bar is true. If one of them is going to be true it is going to be true. So this is first of the Boolean identities if you want to call it that. We will only be using such simplification procedures here and not great things.

Similarly I can have an OR combination of variable and its compliment the output has to be 0 because in AND combination when both the inputs are true only then the output can be true.

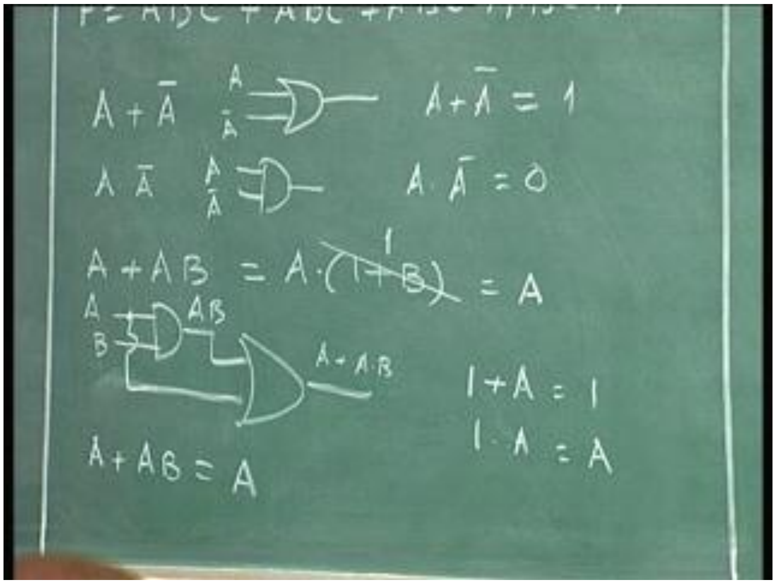
(Refer Slide Time 17:39)



So I am giving the variable and its complement as inputs, one of them has to be necessarily false so the output has to be necessarily false. Most of the time this itself will do for reducing our Boolean expressions because when you have a redundant term a term with A or the term with A bar with a common variable we can take the common variable out and put A plus A bar lock it of as one. Similarly we may have this A and A bar lock it of as 0 most of the time that is good enough. For example, a way of a function A what I mean by this is suppose I have A OR AB that means I have a gate with AB as input feed into a, this is not (Refer Slide Time: 18:31) this is A and B this is A OR A AND B.

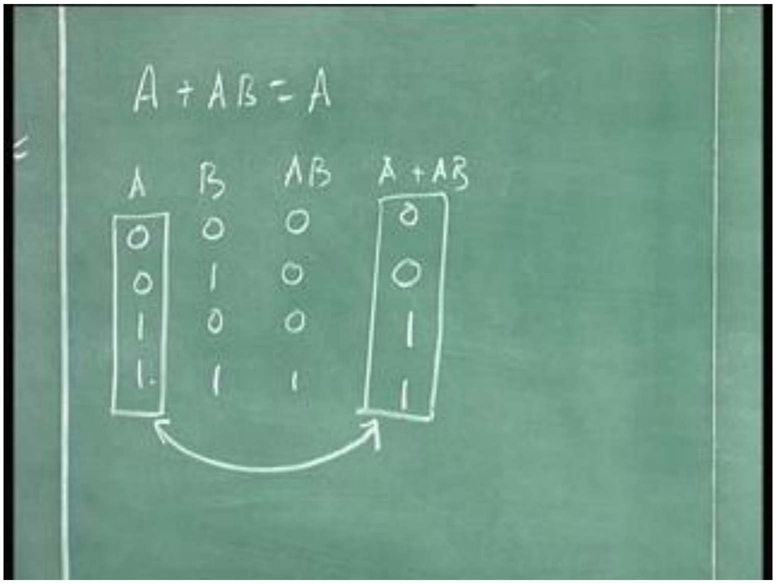
Now use let us use Boolean algebra; A OR AB, what is 1 OR B? Because one is an input which is always true and in OR function if one of the inputs is true the output has to be true so 1 OR B has to be 1. So I can lock it of and say 1 so A AND 1 is A because if A is true function is true if A is false the function is false so this is A. That means A OR AB is A because of the fact 1 OR A is 1, 1 AND A is A if you want to list them all as identities this is one of them, identity, this is another identity, this is another identity, (Refer Slide Time: 19:50) this is another identity, this is another identity. That means if I have a variable present in two terms with extra term then there is no need for this extra term. That means this OR gate is redundant this AND is redundant as if its transmission is A. If A is true this function is true and if A is false this function is false.

(Refer Slide Time 20:28)



If you are not convinced you can always verify it by giving a truth table. I can draw A B AB, I said A OR AB is A I already showed that it can be done by using this relationship 1 OR B is 1, A AND 1 is A but then if you are not convinced you are always write the truth table A B AB A OR AB 0 0, 0 1, 1 0, 1 1, AB is 0 0 0 1 because AND gate and A OR AB is this or this which is 0 this or this is 0, (Refer Slide Time: 21:25) 1 1 so this is same as this.

(Refer Slide Time 22:07)

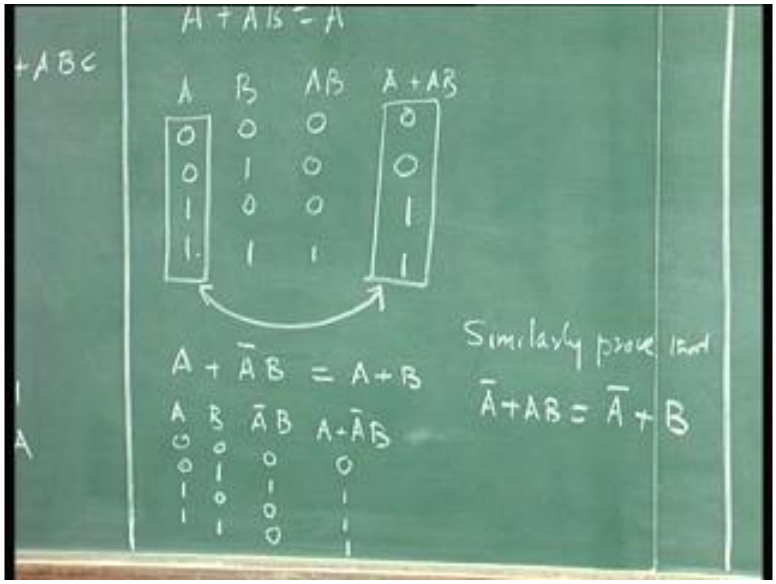


This is one other way of proving Boolean algebra or Boolean identities any Boolean identities. But what I have not mentioned here is all these formulae all these properties of commutation, distribution and all that you are applying to normal algebra. For example, A and B is same as B and A, A or B is same as B or A so this is called commutative law, distributive A OR A AND B OR C is same as AB OR AC it is a distributive law. All these simple rules of algebra used can be used here for example I have used this here. Thus laws of commutation, distribution and all those things are applied here also equally well.

As I said I am not going to do rigorous Boolean algebra treatment that is why I have left out those which are very simple things the things about which we are very familiar I am just skipping so that we can get the essence of this. So what I am saying is by repeated application of these things whenever you find a term like this we can lock it of and get it so you can think of this as a simplification, we will see two other identities and then we will stop this.

For example, suppose we have A what do you think this will be? A OR NOT A AND B. So you can just write this if you want to; A B, this is 0 (Refer Slide Time: 23:22) because of B is 0, this is 1, this has to be 0 because B is 0 and now I am going to combine this with A. suppose I combine this with A OR NOT A AND B then this will be 0 1 1 1 which is the same as A OR B or A plus B if you want to call it A plus B also so this is another simplification possible. You can also prove this by using some of the other identities you have learnt earlier.

(Refer Slide Time 24:54)



Similarly prove that A bar plus AB is equal to A bar plus B. Why am I only choosing those things is because every time we are knocking of a term? Here simplification means reduction in the number of gates or reduction in the number of terms of a product term because the final expression is a sum of product expression, and there are two ways of reducing the logic one is to reduce the number of terms, number of product terms if we reduce you will get a reduction or in each of the product terms if I reduce the number of variables in that term then also there is reduction.

For example, a product term is equivalent to AND gate. There are three inputs, you will have three variables in the product term I need a three input AND gate. If there are only two variables in the product term I only need two inputs AND gate. So you have to reduce in two ways. You have to reduce the number of product term so that the number of inputs with the OR gate will be reduced because some terms fit to an OR gate, the sum of this product term. That means each product term is put into an OR gate OR gate is equivalent to a sum, it is really OR operation but we call it sum just for the sake of understanding.

So how do you reduce the complexity of an OR gate the number of inputs of an OR gate is by reducing the number of product terms so that the total number of terms you need to sum or total number terms for which you need to get an OR function is reduced. And each of these terms which are in AND gate output can be simplified by reducing the number of variables in that AND gate. So, for example if you take this logic function I can reduce this by two ways. There are a total of 1 2 3 4, 5 terms and any number less than 5 is reduction and any number less than 3 in each of the product terms is also reduction. So finally we got this function. This is what we started with. Now I have to show you using the Boolean algebra that we have now developed this it is the same as this. We will do it in a minute.

I am reducing it two ways. I am reducing the number of terms from 5 to 2 and each of these terms reduce from three variables to only one variable in one case and two variables in another case so it is a two step reduction or two way reduction that is why I am giving you the identities that are only relevant from that point of view.

Boolean algebra as I said can be rigorously taught as a tool of simplification of Boolean functions. But in a practical level we are looking at reduction in the hardware both in terms of number of gates and number of inputs to each gate. So all the identities which are relevant in that context namely (Refer Slide Time: 27:58) this is this, this, this, this, this, this, this, this and this all of them are knocking of either a term altogether in this case or in this case I am knocking of a variable and a term, I am knocking of a variable in a term. So I have given you identities which will knock of an entire term or knock of a variable in a given term.

This is the concept of Boolean algebra from the reduction point of view; this is just our primary goal in this course efficient digital design. What is the mantra? It is to reduce the hardware, increase reduction of hardware, cost saving, power saving, reliability increases, size increases or decreases and so on and so forth. So you keep that as a single point goal and from that angle if you look at we are going in the right direction.

Now to quickly tell you that this is the same as this, we will do a quick analysis of this. This is the original function (Refer Slide Time: 29:40) the canonical sum of product written directly from the truth table reading of the entries from the truth table for which the output is true. And I want to make it F is equal to A plus B that is the sum of products this is the canonical sum of products. Any term of this type is called sum of products (Refer Slide Time: 29:53) because it is sum of products term and if it is minimum beyond which it cannot go then it is also called a minimum sum of products. So my goal is to make it as a minimum sum of products. So this is canonical sum of products and by repeated applications we get sum of products which are simpler than this and finally you get a minimum sum of product beyond which you cannot reduce further.

Now how do you do this?

For example, can I use this formula AB bar C bar and C. Supposing I leave this term as it is on a minute C plus C bar is 1. So I am combining these two terms and combining these two terms. We can knock of this C plus C bar as 1, so this becomes (Refer Slide Time: 31:20). Now again I am going to use the same. Now this has to be reduced to this. We can use one of these identities here A and A bar and B. I have an identity here, I have a variable and another term in which the variable is complemented and some other term. So we treat this BC bar as another variable, A appears as A bar here so you have a variable and the same variable repeated in the complement form along with another variable or along with other variables and this extra variable can be knocked off. A OR A bar B is A OR B.

That means A bar gets reduced, and here also the A bar will get reduced. So it is the same as this (Refer Slide Time: 32:58), so if you are not convinced there are other methods of doing it. After all as I said there are several ways in which you can do it. That is the problem with the Boolean algebra. Y it has to strike you to apply the right identity at the right place and you never know whether it can be simplified any further, nobody is going to tell you this will be the simplest.

In exams of course they will tell you, prove that lhs is equal to rhs that is a trick, there is also a trap there. You somehow start with lhs go half way start with rhs go half way and somehow make a big confusion, make so many terms and strike them off, remove them, erase them and say therefore lhs is equal to rhs so I don't know but that is one way of looking at it. But in a rigorous way it is a question of whether you are trying to do it for marks or trying to do it for a design for living, if you are trying to do a design for living then make any efficient circuit and then we will not do all those dirty tricks. Then you have to make sure that you have got the simplest possible circuit.

So if you don't like this application here because it may not strike you here what I can do is I can always combine this, I am going to add one term here, I am going to reuse this term (Refer Slide Time: 34:40). I have already used this term A B bar C here and simplified it here. I am going to reuse this term here the third term A B bar C I am going to reuse it. Now if I reuse it I am going to combine these two, A I leave, these two I will combine as this (Refer Slide Time: 35:23) this is also true, I have not done anything different.

(Refer Slide Time 37:14)

I should have combined the fourth term. I should not have combined this term the fourth term which I have already combined once AB C C bar ABC, I am reusing this fourth term so it becomes A B C bar and take B C bar out and write it as A plus A bar and lock it of make it one is it valid is it legal? Yes it is, because what is A OR A? I can have an OR gate with the same input tied to all the inputs and still the output is A. A OR A is A, what is A AND A? A. This is saying if I have an OR gate if A is 1 both the inputs are 1 and the output is 1 and if A is 0 both the inputs are 0 and output is 0. Similarly in AND gate if A is 0 this also is 0, this is 0, this is 1, this is 1, this is also 1. That means I can use repeatedly the same term again and again for simplification purposes that is another way of doing it.

So I have two methods here at this time. I had this of course all this time I knew that there is a simpler function which is A OR B AND C bar since I knew that I was going to try. Supposing I give you this problem in the exam up to this may probably do by combining this A plus A bar first step clear, clear here (Refer Slide Time: 37:37). Here this is the way the Boolean algebra's problem comes in from the practical point of view. You may stop here and say you will argue with me that after all I got this you have to give me full marks, somebody will say reduce 1 mark, this is not a question of mark as I said again.

Therefore it is problem solving for practical use. So here it is difficult for you to strike that it can be reduced further by looking at this an identity corresponding to this or if you don't strike this if it doesn't strike you then you have to write it this way but what if you don't write it that way then it's okay. What I am saying is this way you are not sure of simplified process. Because now I had a goal I wanted to show it to you that is possible I tried to put it this way.

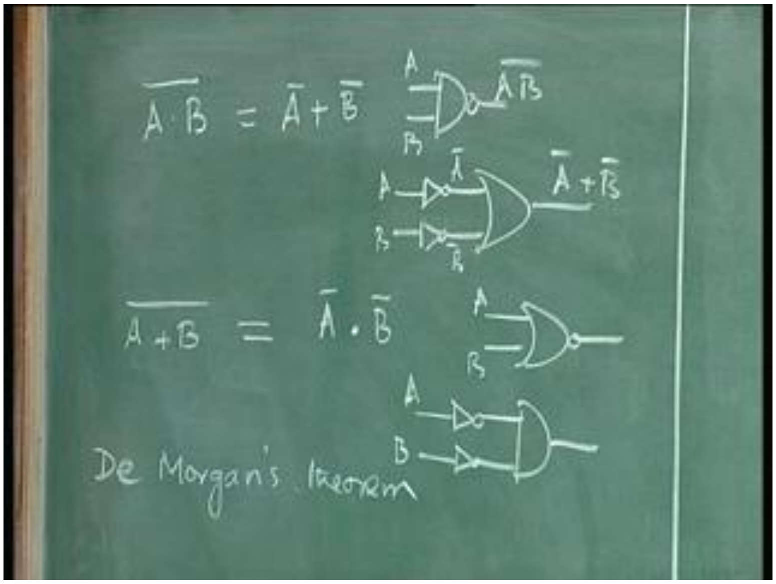
If I didn't give you this F is equal to A plus B B C bar the easy output of this is the minimum possible sum. We would probably stop here this is okay, there is nothing wrong, I am just saying. So Boolean algebra tool is very useful but you have to be careful you should remember to apply correct identify in the correct place and be sure when to stop beyond which you cannot reduce further. But we proved that this is possible now, this is Boolean algebra tool.

Now, again this is a minimum sum of products, from canonical sum of products we came to sum of products. Any of these expressions in between sum of products are not canonical, not minimum. This is not the original line, this is not the final line, it is the sum of products because this looks like a product terms summed into a series. Finally you have this which is the minimum sum of product. And each of these terms for which the output is true from a truth table is called min term. There are eight min terms out of which five min terms have output as true and three min terms have output as false. So we talked about truth table, canonical, sum of products, min terms, sum of products and minimum sum of products.

Now an equivalent approach can be done from a slightly different angle that is called De Morgan's theorem. Again this is some thing which you might have learnt even in high school in your set theory if you remember. Do you remember set theory? Some of these things you learnt in set theory. I will quickly state De Morgan's theorem and it can also be done in a slightly different way using De Morgan's theorem.

Suppose I have a NAND function and complement the overall instead of complementing each term, I convert each variable into its complement and convert the operation into its complement. OR is considered to be a complement operation of AND. So AND operation is converted into OR, A converted into A bar and B converted into B bar. So if you have a function AB A AND B the complement of that function can be written as OR operation of complement of A and complement of B. This we already proved when I tried to get this NAND gate concept. What is A AND B? This is only a NAND operation. Did you not say this is the same as inverting the inputs and OR in them. Likewise I can start with an OR operation and complement the whole way write it as OR complemented as complement of AND operation, A complemented of A bar B complemented of B bar. Again saying the same way that I can have a NOR gate function this is equivalent to complementing A and complementing B and feeding to AND gate. These two identities are called De Morgan's identities or theorems.

(Refer Slide Time 42:51)



Many times the De Morgan's theorem is used. So if you are not clear about that F is equal to A plus BC bar that is for original function we talked about you are only given the truth table, you are only given the min terms for which the output is true five min terms for which the output is true, the original truth table or the canonical sum of product expression in terms of min terms. There is another way of approaching because there are five terms for which the output is true; I want to simplify it to a minimum number.

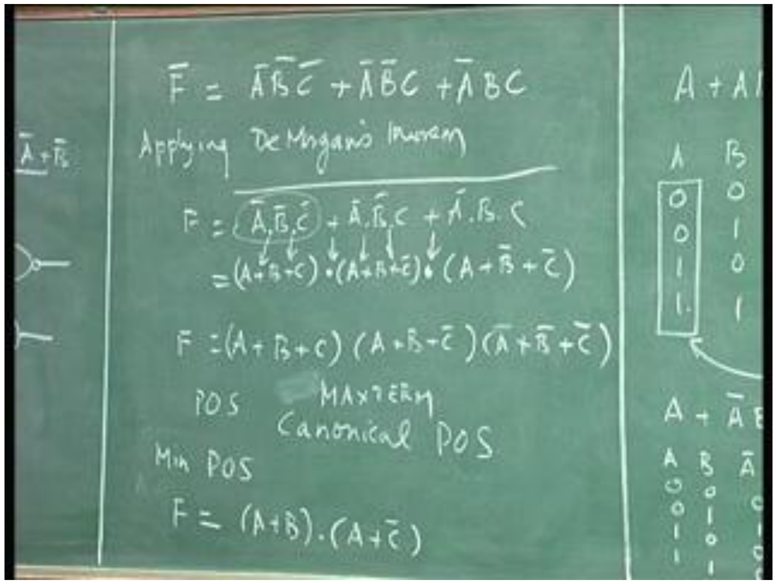
Can I see which of the terms for which the output is NOT true output is false; there are only three of them. Out of eight possibilities for five terms it was true and for these three terms it was NOT true so is it possible probably to start with those three terms and simplify it may be instead of simplifying for five terms and a smaller number? May be it is possible. Perhaps it is possible to start with three terms and simplify it. This approach can be tried so the same truth table if you take F is NOT true for the three possibilities. If you remember the original truth table I will not write it now again. For the first row for the second row and for the fourth row output is false. The original truth table row 1, row 2 and row 4. So what is it for row 1? It is A bar B bar C bar for which the output was false so F is NOT true for this combination, NOT true for this combination and NOT true for this. These are the three combinations for which the output is NOT true.

Now I am going to apply my De Morgan's theorem in that result. If I have a function its complement is applied by De Morgan's theorem on each of the terms on the right hand side. So from here applying De Morgan's F can be written as this (Refer Slide Time: 45:47) which is same as, this operation becomes AND operation and this operation becomes AND operation and each of these AND operation becomes OR operation, each of this AND becomes OR and each of these variables get complemented. That means it becomes A OR B OR C, this is AND complement removed, complement removed, and complement removed and (Refer Slide Time: 46:23) and this OR becomes AND C becomes complemented, B gets complemented, complement of B and complement of A get removed and finally you have C bar.

I will rewrite it here to avoid confusion. Each of these terms is called a sum term because it is a sum operation and this is called a product of sums. So this is a product of sum F is POS product of sum, each of this is a sum term and multiplication of the sum terms gives the final expression so it is the product of sums and it is also called MAX terms. MAX term is what you got directly from the truth table. Min term when you look at the ABC type of thing, MAX term is when you look at A OR B OR C so OR operation if you apply it is called MAX term.

So MAX term is of all the terms of the truth table, eight terms of the truth table and each of them can be written as a MAX term. For some of the MAX terms the output is true, some of the MAX terms the output is false. And product of sum expression is written using the MAX terms for which the output is false. You write down the MAX terms for which the output is false, you get a product of sum expression this is also called as canonical product of sum expression because canonical product of sum so I can simplify this further by a repeated application of Boolean algebra to give me the minimum product of sum minimum POS like minimum um of product. So I can start with the truth table proceed with min term canonical sum of products, reduce it to a minimum sum of products. Or given a truth table I can start with max terms, product terms, write a canonical product term expression for the output and reduce it further to get a minimum product of sum expression.

(Refer Slide Time 50:42)



If you do that by just multiplying this and by repeated application of Boolean algebra you will get the expression as, I will give you the result, I want you to try it as an homework F becomes A OR B I want it in this form, I can always apply Boolean algebra, I said F is equal to A plus BC bar don't proceed from here and prove this.

I wanted to go through this exercise of taking the canonical product of sum expression from the truth table because I started with something which is true instead of starting something with true I started with something as false applied De Morgan's that means this becomes something for which it is true so this is the function. But the expression is not sum of product it is the product of sum expression. I wanted to apply to this product of sum expression our Boolean identities known Boolean identities or unknown Boolean identities any identity which I would have missed here is also fine with me and get it in a minimum product of sum and that happens to be in this form, in this form I want.

So can you do that as an exercise that is A OR B and A OR C bar. So now I gave you a flavor because I don't want to spend too much time on this Boolean algebra. But at the same time you should know that it is a very powerful tool you used, very easy really what is the big deal about it?

It is just knocking of the variable wherever possible to reduce the complexity. You can lock of terms, reduce the input to the OR gate, lock of variables to reduce the input of the AND gate that is the crux of the Boolean algebra. We did that, it is not all that complicated, the only thing is uncertainty. The right identity you should use if it doesn't strike what happens and you never know when to stop unless you are given what is the minimum sum because you are the designer nobody is going to tell you this is the minimum if it is an exam I can tell you reduce to this form, if it doesn't come to that form then you know you have not got the result. But for that it is a good tool but as I said we will remove the uncertainty by applying the graphical method the map method which we will see in the next lecture.