Digital Circuits and Systems

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

Lecture - 6

Karnaugh Maps and Implicants

Today we will continue with the simplification of Boolean functions or logic functions. The last lecture we talked about Boolean algebra being a set of formulae or identities to be used to simplify a given logic expression without changing the Boolean relationship. That is for a given set of input conditions an output is defined as true or false so when we make a simplification that should not change. Whatever simplifications you make the output should still be true for all the input combinations for which it is supposed to be true or it would still be false for all input combinations for which the output is supposed to be false.

Now we will continue this and use a graphical method. As I mentioned earlier it is not a new theory or anything. It is a systematic procedure actually. If you want to call it as a procedure at best it is a procedure, systematized for easy handling or when the function gets bigger and bigger so that it can be automate.

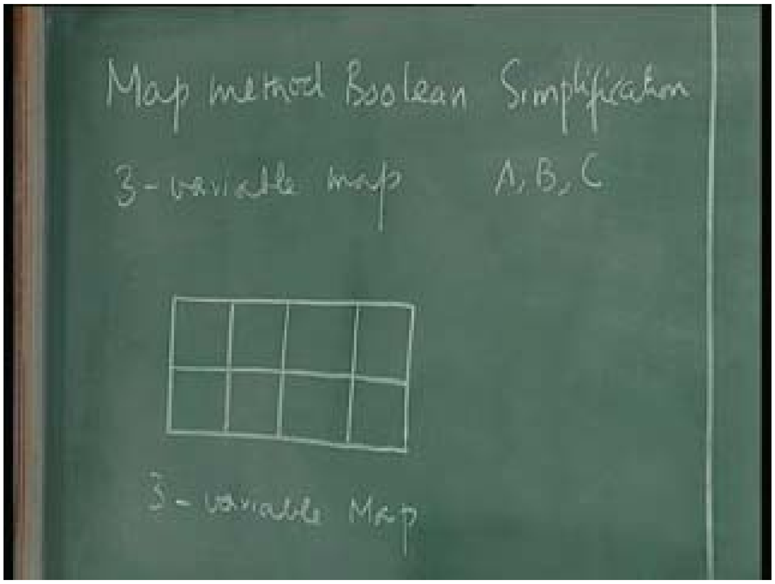
There are other automated reduction techniques other than graph method. We will not see them in this course. But these are all having the same type of concept. Identify wherever possible the combining of the terms. What did you do yesterday? In order to reduce or simplify the function if you can reduce the number of terms it is good and in each term if you can reduce the number of variables it is good. So we will use the same method to see wherever possible we can combine the terms into a smaller term. Two terms can be combined into a smaller term then there is a saving in hardware. The same concept will be applied in graph method and later on there are methods called as computerized methods, implicant methods and all that. We will not see in those things because conceptually they are same, they are merely procedures. If you know how to do it conceptually you can learn the procedure any time depending on your requirement.

So we will start with the map method of Boolean simplification today. It can also be called a graphical method but usually it is called a map method. What we have to do is to map the truth table on a graph. Look at the truth table there are 1s and 0s in the output and we want to combine all these 1s together to give the minimum possible solution. Now we will map it on a graph and see whether there is a possibility of combining those that’s all we are going to do.

We are going to repeatedly use the same concept of Boolean algebra again and again. There is nothing like we are going to use a different set of equations or identities, it is the same set of equations or identities we are going to use but in a systematic way.
So let us take the example; what is a map? A map is nothing but all the variables are present in a map in true form and complement form. In other words we will have a value

corresponding to each input combination that is possible. Let us take an example of 3 variables. Let us say the variables are A, B, C, what are the eight combinations of A B C? They are A bar B bar C bar; A bar B bar C etc finally ABC. That means there are eight possible values. So we will now draw a graph with eight cells. This is how it is going to look like. You can draw it horizontally or vertically doesn't matter.

(Refer Slide Time 7:18)

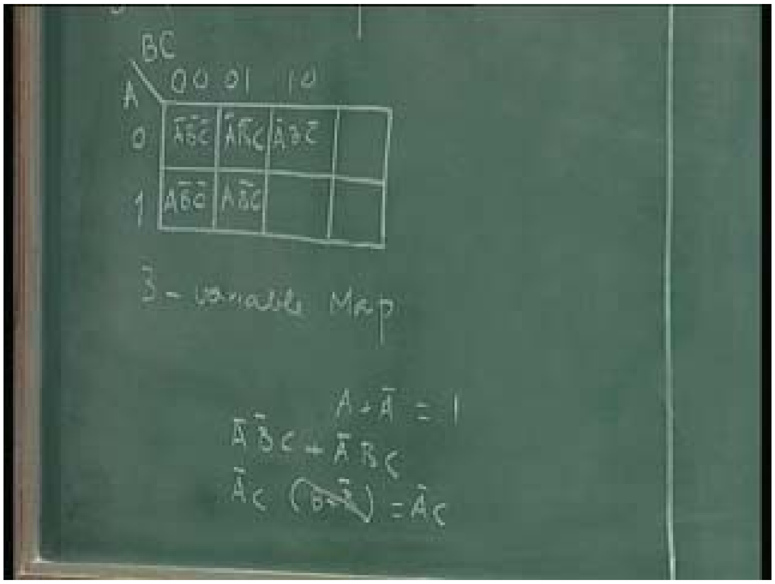


I need eight cells in this map I call this a 3 variable map, eight cells corresponding to this A bar B bar C bar; A bar B bar C; A bar B C bar; A bar B C etc will be represented here. I will say on this vertical axis I will let A vary the true complement value, true value. this 0 here means A bar this cell will have a value, the value of this cell A is A bar, value of A for this cell is A bar and value for this cell and for the horizontal axis we will let B and C vary there are four possibilities of B and C varying such as B bar C bar, B bar C, B C bar, BC so that can be represented by 0 0, 0 1, 1 0, 1 1.

When I say 0 0 here for BC, A0 B0 C0 correspond to A bar B bar C bar that's all. Here this will be A B bar C bar you don't have to write it, it is the first time I am doing so I am writing the corresponding values in the input combinations, input combination I am writing in the cell but the cell is supposed to have that combination. Once you are familiar with this you don't have to write it. Then the next will be 0 1 for B and C so this will be A bar B bar C, this will be A B bar C. So what will be the next value for BC? It is 1 0 or 1 1 but if we put 1 0 what will happen is I have to write A bar B C bar. But I would like to have this adjacency rule. Remember, this type of identity I am used to, repeatedly to combine to knock of a variable.

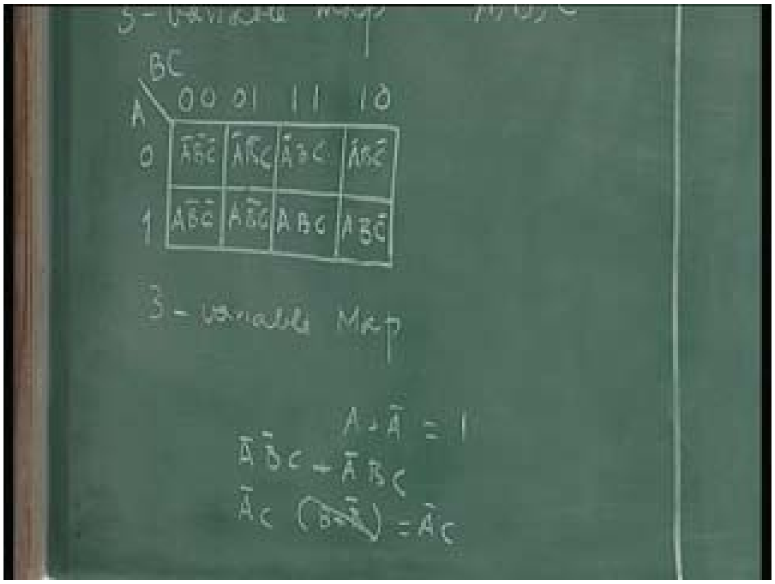
For example, if we have A bar B bar C OR A bar B C then I can take this as A bar C and B or B bar and knock it of and write it as A bar C. that means I should only let one of these two variables change. Between B and C I should let only one of those variables change from cell to cell. From one cell to the next cell either in the horizontal direction or in the vertical direction if you let only one variable change and the other variable keeps its value then I will be able to combine it with a previous cell.

(Refer Slide Time 10:08)



In this case when I say 0 1 and 1 0 B changes from 0 to 1, C changes from 1 to 0 both the variables B and C change at the same time that means it is not possible to combine them effectively. So the technique is then not to write here for 1 0 for BC write here as 1 1. That means I will let B change from 0 to 1 and C remains and I let C remain as 1. That means this will be A bar BC and this will be ABC. And finally the only other value which you have not used is 1 0 that means B is 1, C is 0, A bar B C bar A bar A B C bar.

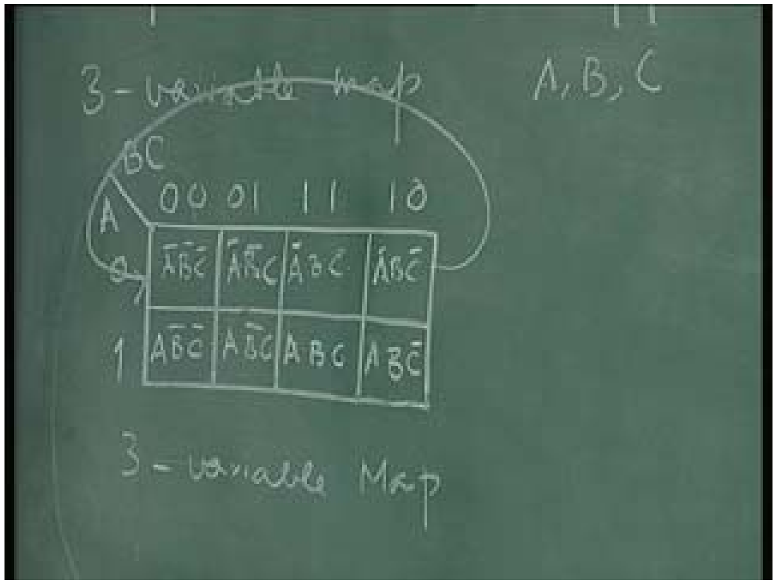
(Refer Slide Time 11:53)



The advantage of this representation is between two adjacent cells either in this direction or in this direction only one of the variables changes from 0 to 1 or 1 to 0 the other variables remains same that means I can use this formula repeatedly. This formula identity A OR A bar is 1 can be used if two adjacent cells are same and both of them are one I can combine them to knock of the variable which varies from this to this and retain the other two variables which do not vary from (Refer Slide Time: 11:40) this cell to this cell.

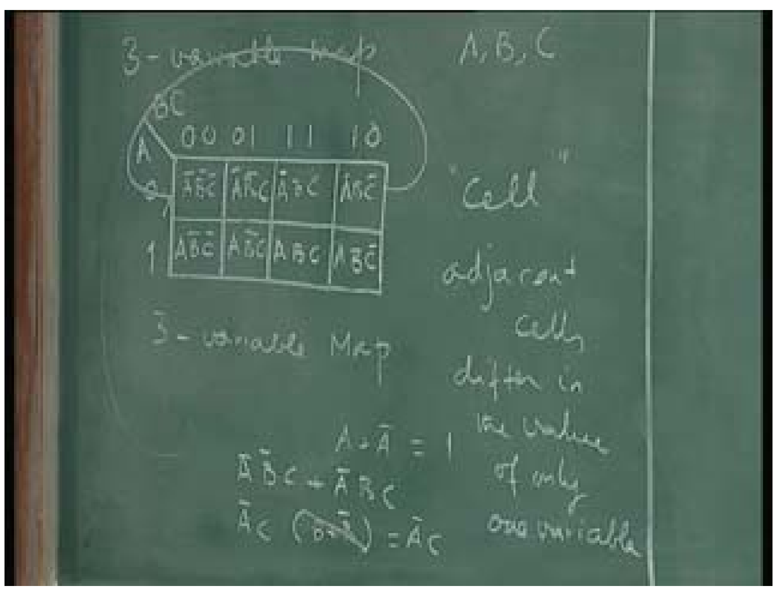
The same thing applies here also. here A only changes, B value C value remains same in all these cases and this adjacency work also backwards round its a wrapping it is something like a circular symmetry here so between this cell and this cell A bar B B bar C bar C bar that means B only varies, B is 0 here 1 here C is 0 0, A is 0 in both cases. And if I had four columns or four rows in the vertical direction then also I can apply rotational symmetry if I had two more rows this and this will be adjacent.

(Refer Slide Time 12:30)



So in this map I will call each of them as a cell and the cells in the map which are adjacent to each other in horizontal direction or vertical direction will differ in only one variable and all other variables will be same these are called adjacent cells, differ in the value of only one variable.

(Refer Slide Time 13:16)

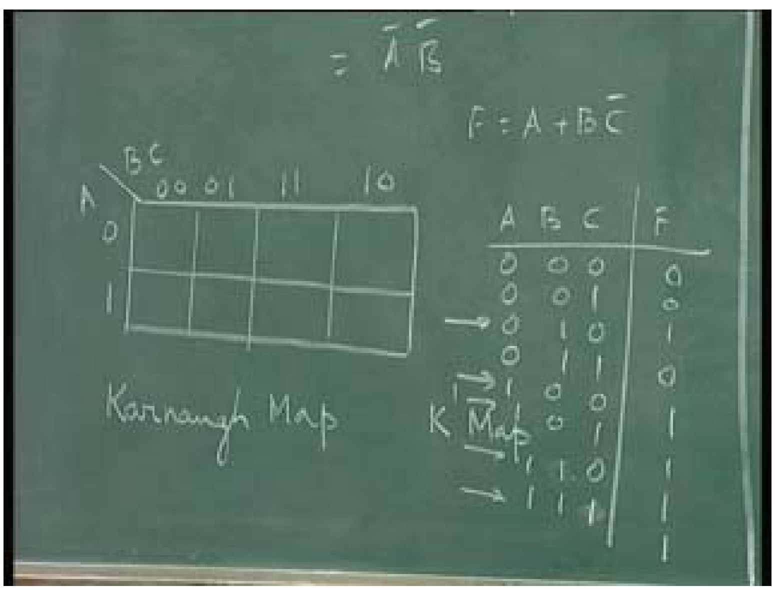


Adjacency applies from end to end also that is what I am trying to say. I am trying to say that adjacency applies from end to end as if we are folding it around wrapping it around in a circular way. So what does it mean now? If I have values of output is 1 or true for this case and for this case that means the truth table will have a 1 in this cell as well as in this cell when I have a 1 and 1 in two adjacent cells of the truth table I can knock of the variable which is different from these two cells and have a simplified expression, that is what we did. For example, yesterday we said we had A bar B bar C bar AND A bar B bar C both of them are one in my truth table and I have written it as (Refer Slide Time: 14:32) so that is what we are going to use in a graphical method.

We are going to use the identity A plus A bar is equal to 1 repeatedly in the graphical method till we can identify the entire group of 1s till we can do it no more. Hence it is a more systematic way of doing it and you more or less certain at the end to be exhausted all the possibilities because it is a graphical representation. So let us map our original function which you have been talking about in the last couple of classes. This is how (Refer Slide Time: 15:15) you write the map you always write the map like this with all the cells and mark with 0s and 1s on the left hand side for this variable and 0s and 1s on the top for these variables then only the map is complete. This map is also called as Karnaugh Map. Karnaugh is the guy who probably invented this, it is named after him, they simply call it K' Map. You will see this K' Map mentioned in text books or Karnaugh Maps.

Let us apply that function in the truth table that we had originally for F is equal to A plus B C bar. Which are the entries of the truth table? It was 1 in the output. You remember ABC this was the truth table, this was 1 BC bar. This was the original truth table (Refer Slide Time: 17:04) we have been talking about in the last few classes, the examples.

(Refer Slide Time 17:25)



So there is a 1 corresponding to this row and all these four rows the output and for all other rows the output is 0. So the output was 0 for the first row, second row and the fourth row, 1 for third row, fifth sixth seventh and eighth rows but we don't call it 1 to 8 we always call from 0 to 7, the reason is very simple, with 3 variables I can have eight values but if 0 has been included as one of the values I can go only from 0 to 7 or eight values.

So when you have a number representation 0 is an essential thing I can't have a number representation where I cannot represent 0. So with the binary number one with one digit or one bit of binary even though there are two values possible the one of them happens to be 0 or the other one happens to be 1 so I can only represent a maximum of 1 using a binary bit.

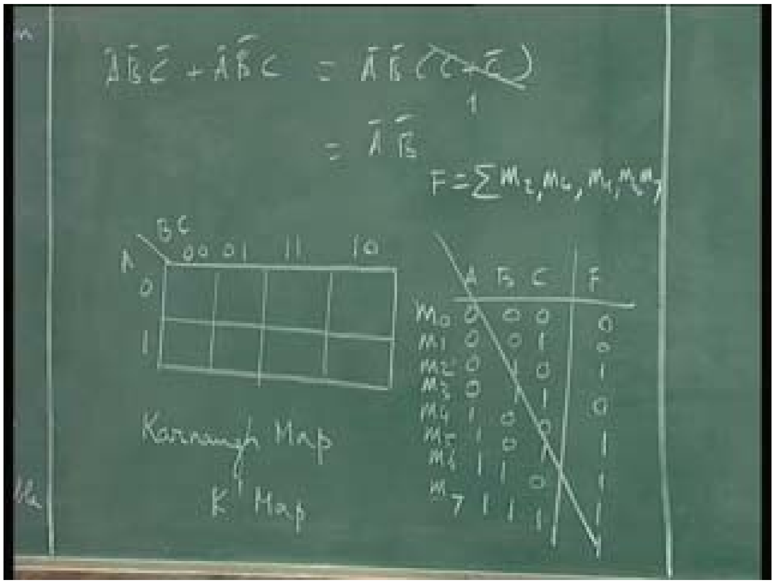
We have two binary bits or two bits I can have four values but the first value is 0 so it is 0 1 2 3. 0 is 0 0, 1 is 0 1, 2 is 1 0, 3 is 1 1, I can't represent four using two bits. Since it is going to be binary and we always have to have 0 as the one of the values in digital design or digital system or digital representation wherever digital things are involved we always start with 0 and not 1. So the first entry is always a 0 entry so that way I will call this zeroth row, first row, second row, third row, fourth row, fifth row, sixth row, seventh row and we also call these variations or combinations of the inputs each of the input combinations of a truth table are called min terms. So first min term is min term 0, second min term is min term 1 so there is a symbol for the min term a small 'm'.

Already there is K' Map here so this is called min term 0, min term 1, min term 2, min term 3, min term 4, min term 5 so min term 0 to 7 are the eight min terms possible for with 3 inputs and when it comes to max terms we talked about max terms also yesterday we use a capital 'M', a capital 'M' for max small 'm' for min. so I don't even have to write this truth table. Of course this one is what we should get finally this is what we started with. But this is the truth table which we considered and said that this is same as this. We first do a circuit for this then you said that this circuit can represent a truth table of this type and then we proved yesterday using Boolean algebra that this table can be represented to this function.

Now my objective is to reduce this table again to this using Karnaugh Map. Yesterday we did it using Boolean algebra today we will do it using Karnaugh Map. So I don't need this any more unless at the end to verify whether the result is right or wrong. This is what I should get. We will see whether we get that.

So I have the truth table and I have to represent it in Karnaugh Map, map it in a graphical representation in a K' Map and you don't even need this truth table. Somebody is going to give you the truth table and then you are going to map it So why waste time because you know what the truth table is all about all possible input combinations and the combinations for the input for which the output is 1 and this information is called the truth table. You list all the input combinations possibilities and also list those combinations for which the output is 1.

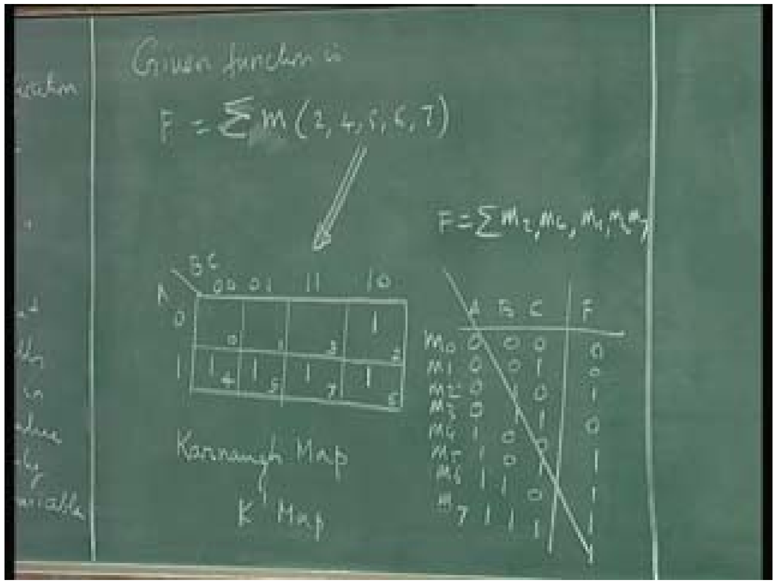
(Refer Slide Time 22:25)



Therefore instead of saying all these in such a big way can I not simply say this function F has output 1 for m_2 m_3 m_4 m_5 m_6 m_7 because you know what is m_0 for 3 variables it has to be A bar B bar C bar or P bar Q bar R bar or whatever X bar Y bar Z bar whatever. It can be X Y Z, P Q R, alpha beta gamma, A B C, doesn't matter. but as long as there are three variables, only these eight combinations are possible so what is the point in writing everything in a big table and say this is 1, this is 1 and this is 1 instead I am saying this F is equal to min term 1 m_2 min term m_4 min term $_{m5}$ min term m_6 min term m_7. For these min terms output is true or 1.

Since it is a sum of product expression if you remember each of these product term I represent it by a sigma. So we now say my function is I will remove this now (Refer Slide Time: 22:30) the given function is F is equal to sigma m standing min terms instead of even having to say m m m repeatedly try to minimize having to repeatedly do the same thing over and over again out of laziness or whatever you want to call it out you want to say stationery or just more or less boring just don't have to keep on removing the same thing over and over again so what I am saying is sigma m means the sum of the products and what are the product terms involved or the min terms? It is 2, 4, 5, 6, 7. So if I give you the function like this F is equal to sigma m2 four five six seven you know the truth table and you know the entries in the truth table for which the outputs are 1 and the others are 0.

(Refer Slide Time 24:57)



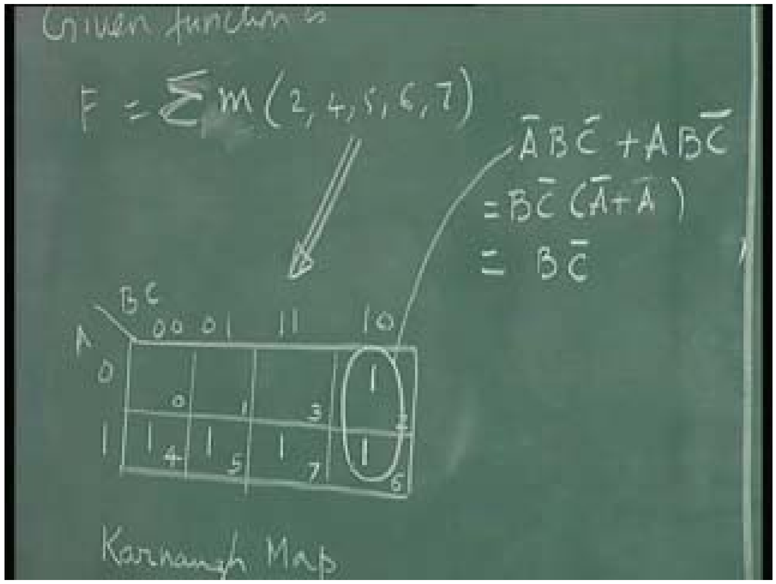
So my job is to map this into the Karnaugh Map represent into this Karnaugh Map this truth table so I don't need to give you this so from here it should be able to directly come to this you would directly be able to get this.

Now, in order to help you in the beginning, later on you may drop this, you can write these min term numbers in the cells. This is min term number 0, 1, 2, 3, 4, 5 so that you won't make a mistake in the beginning and then later on you know all that by experience. These are the min terms my job is to map or represent on this map the min terms for which the output is 1 or true these are 2 4 5 6 7. So this is the K map of the function given as this which can also be expressed as this or can be expressed as sum of products as A bar B C bar plus A B bar C bar plus plus plus five terms we wrote yesterday.

So this is the Karnaugh Map and this is the function how do you simplify it? Deduce it so that fewer literals are used and fewer terms are used in final implementation in the final representation. that is where adjacency rule comes in. you know the two adjacent cells differ in only one variable so you find a one in one variable one cell and one in the adjacent variable cell, adjacency works left to right, top to bottom, end to end not diagonally. Adjacency does not work diagonally you have to remember that. There is no adjacency diagonally.

So anywhere I find in a 1 top to bottom or side to side I will try to combine these two and find out which is the variable which is appearing in one cell as a complement and the next cell as a true. In one cell it will appear as B bar and in the next cell it appears as B then B goes because of the A plus A bar is equal to 1 or B plus B bar is equal to 1 relationship. That is easy to identify for example these two cells A appears as 0 here, 1 here. When you take these two cells and connect it like this, this is equal to, later on you will be read it directly from the map. Now the first time I am going to write it as this is A bar B C bar this corresponds to B, (Refer Slide Time: 27:06) this corresponds to C bar and this corresponds to A bar. So you write it as A bar B B C bar and this corresponds to A B C bar. These two together can be written like this.

(Refer Slide Time 28:47)

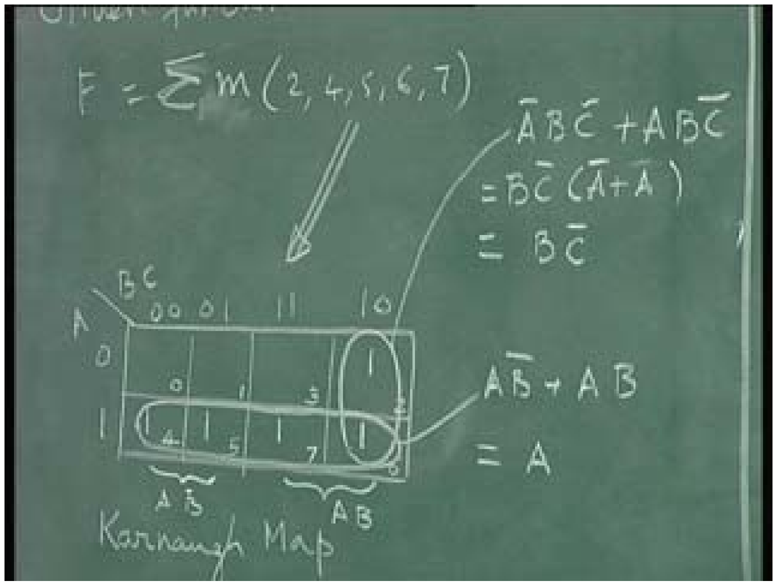


What is common between these two? B is common and C bar is common so B C bar is common so take BC bar using that distributive property say A plus A bar is 1 and since A plus A bar is 1 it is a known identity this is same as this. So these two cells can be merged or combined into a single term of BC bar. Each of the cells have three terms three variables I have knocked of one of the variables so two 3 variable terms have been reduced to one 2 variable terms that is the simplification procedure.

There is nothing very great about it, all you have done is identified the adjacent cells where 1s are marked and have found out which of the variables was (constan….28:35) these two cells which of the variables change between these two cells and the changing variable is removed, that is all. Hence you don't need all these algebra because you know B is constant between these two, C bar is constant between these two, A goes from 0 to 1 so since A goes from 0 to 1 you can write this without A with B and C bar. So you don't need this algebra I just wrote it for your convenience.

Now let us extend this argument to these two and these two (Refer Slide Time: 29:15). These two will have A common, B common, C varying so I can write this as AB. Between these two A is common, B bar is common, C changes so these two if I write together it will be A B bar because A is common between these two B bar is common between these two and C changes from here to here which has to be knocked off. and between these two A is common, B is common in these two, C changes from here to here so C gets knocked off so this is AB. But again these two cells and these two cells adjacent here it is B bar, here it is B, A remains constant throughout so instead of doing it in two steps and instead of grouping two 1s each time and then combining them again with one more step I can group also four 1s at the same time. These two can be combined to as the one group in which it is A B bar OR AB which is equal to A. I don’t have to do it this way step by step but I just showed it to you how it works. But you don’t have to do it this way.

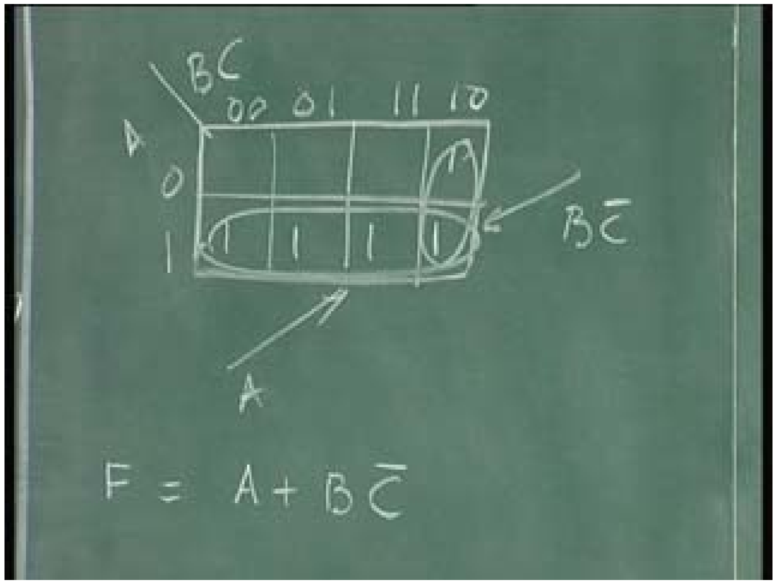
(Refer Slide Time 31:51)



All you have to do is see that A is constant throughout these four cells across the four cells A is common, B changes from 0 to 1 across, C changes from 0 to 1 1 to 0 so both these variables B and C or neither of these variables B and C remain constant throughout so they both get knocked off. So it is not necessary to knock off only one variable, you can knock off two variables, you can knock off three variables and in order to knock off three variables you need eight 1s in a group.

Therefore if you have a more number of 1s in a group adjacent cells it is better for us because we are going to simplify better, our reduction is going to be more. That means without going through all these algebra all this background theory if I give you this same map now all I said was 1 1 1 1 1 I will simply do it combine this and combine this (Refer Slide Time: 32:08) this combination I will write it as A this combination I will write it as BC bar so my final function is the OR combination of this or this BC bar. This is the function we started with that means K’ Map can also give you the simplification that the Boolean algebra gives. But here you know that I have exhausted all 1s, how do you know it is done? I have exhausted grouping all 1s no individual 1 stays out of course if it stays out you have to write it separately and we have also always considered the largest possible groups of 1.

(Refer Slide Time 33:12)



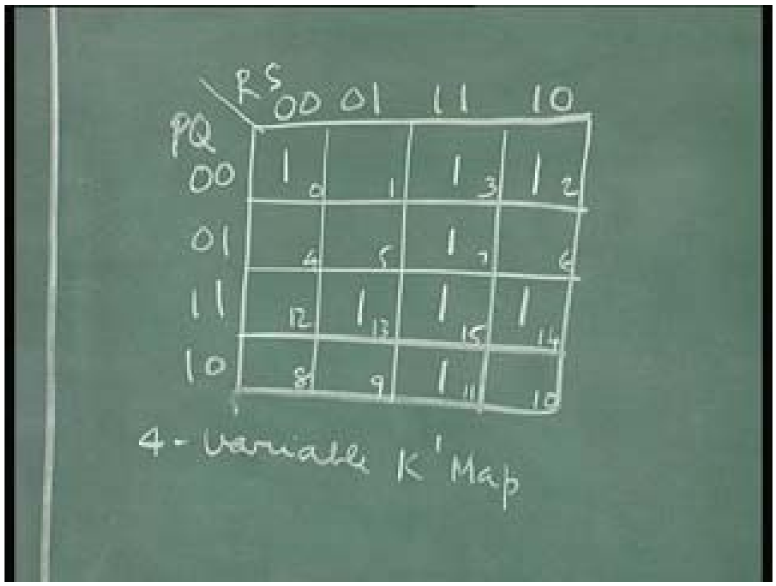
There are four 1s possible plus here there is a group of two 1s possible so whereever possible you have taken the largest possible group in that particular group so there is no question of doing it better than this. So this way you are more confident of your final output compared to Boolean algebra where you might have or might not have used an identity which should have been used. there lies the difference between a Karnaugh Map method and the Boolean algebra but the Boolean algebra is what we have used finally A plus A bar is 1 is Boolean algebra. It is not as if some new technique I will not say new theory it is a technique, the procedure may be different but not the theory. And now you can extend this to four variables. Four variables will have sixteen cells, two variables here two variables vertically and two variables horizontally and you can do like that.

We will take an example of four variables. Simplify minimize using K' Map the function F I can now give the variables. For a change we will go from ABC P, Q, R, S, function of four variables P, Q, R, S which is true for the following min terms. I am going to write arbitrarily I am not having any particular function in mind. When you put sigma m that is the following min terms are having 1 as the output or true as the output and rest of the min terms are having 0 as the output that's what we write. This is a sum of the product representation in min term form.

So let us say 0 2 3 7 11 13 14 15 and so on. It is an arbitrary example. No specific system I have in mind no specific circuit I have in mind. so I have a function of PQRS four variables and if I write a truth table I will have sixteen rows and to this sixteen rows min terms min term number 0, min term number 2, 3, 7, 11, 13, 14, 15 I have true outputs or one output and other min terms namely 1, 4, 5, 6, 8, 9, 0, 12 I have 0 or false as outputs. For such a function I wanted to do a Karnaugh Map and minimize it, reading it by combining the ones as effectively as possible write the minimum possible expression minimum sum of product MSP not sum of product expression SOP is sum of product but this is non minimum. So it is a canonical sum of products which is the standard truth table representation then the sum of product is anything which is having some product relationship and minimum sum of product is what you cannot reduce further as a sum of product expression.

Let us do the Karnaugh Map, we will have four variables P, Q, R, S sixteen cells must be there, this is the K' Map of the 4 variable 0 0, 0 1 remember 1 1 and then 1 0 because the same argument we had earlier for BC and the horizontal scale we will now apply to PQ the vertical scale.

(Refer Slide Time 39:46)



I want only one variable to change from here to here so only one variable changes. If I had put 1 0 both the variables would have changed I don't want that situation then they are no more adjacent. Two cells are defined to be adjacent only if they differ in one variable and if they differ in more than one variable then they are not adjacent according to the adjacency relationship. Even though physically you put them together they don't become adjacent. So this is min term 0 1 2 3 min term 4 5 6 7 min term 8 will start here because there is 1 1 that comes later than 1 0 so 1 0 will have a min term 8 9 10 11 12 13 14 15. this is the 4 variable Karnaugh Map which is standard whether variables are A, B, C, D or P, Q, R, S or X, Y, Z, W or whatever alpha, beta, gamma, delta.

Now what are the min terms for which the output is true or 1 those have to be mapped on to this graph? These are 0 2 3 7 11 13 14 15. Later on when you start doing a few more of these exercises you will drop these min term numbers so this will become less cluttered from a clearer map. Now I put it purposely for you to get used to this so you can remove all these and later on you will not see those.

Now we have to combine them try to group them as many 1s as possible a in a given group but they should all be adjacent. I can straight away see a group of four 1s 1, 2, 3 etc. You always start with the largest possible 1s largest possible group and then come down, don’t start with the smallest number because sometimes the smaller group may get submerged into a larger group later on so start with a larger group.

The idea is to combine all the 1s but you can combine a 1 more than once. remember the Boolean algebra, yesterday when we tried to simplify the Boolean algebra we used the same time twice because A AND A is A, A OR A is A so I can use the same term twice no problem but I should not leave out anything everything should be included but it is alright to include a particular cell which is asserted as one more than once doesn’t harm us. In fact it will help us sometime.

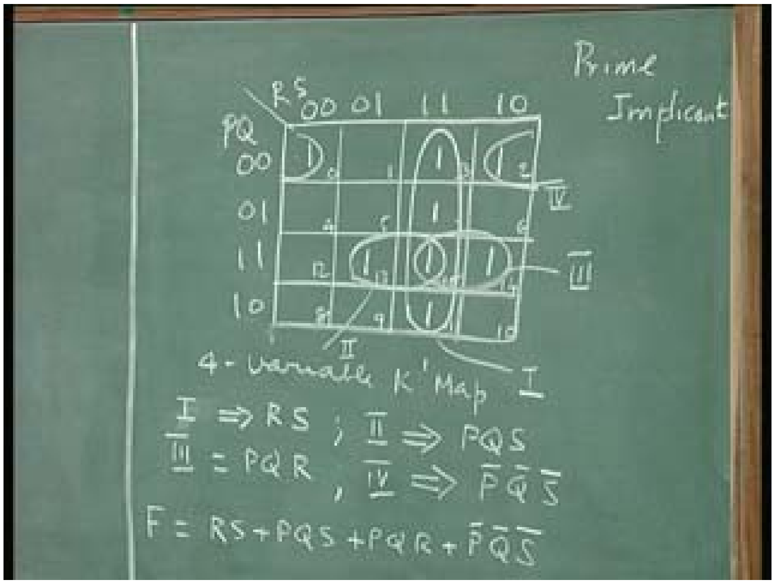
So first group of 1s is this (Refer Slide Time: 41:00). I will call this 1 because there is no space here I will write later on. then we have these two 1s, these two 1s, these two 1s and so on so there are four 1s available here and all of them are to be combined I can’t see this one being combined anyway other than this because all other adjacent cells are blank so this has to be combined this way so is the case here. here though I don’t have to combine in this way because this cell is adjacent to this cell because this is 0 0 0 0 this is 0 0 1 0, R changes from 0 to 1 between these two cells 0 and 2 R changes from 0 to 1 whereas PQ and S are remaining as 0 0 0.

So cell number 0 and cell number 2 are adjacent even though they are physically not adjacent. That is what I am saying physical adjacency is different from logical adjacency we are talking about here. We are talking about logical adjacency. So that way I don’t have to mix this here since this is going to be combined this is going to be combined, (Refer Slide time: 42:25) this is not being combined anyway so I will combine these two so I will call this 2, I will call this 3 there are four terms.

There are now four terms of groups of 1s and each of them I have to write a sum of product expression then my final output F is sum of all these product expressions SOP minimum sum of product. So what is 1? 1 is this, variable P AND Q change all over so you have to write only this R AND S are 1 so RS. So 1 corresponds to RS, 1 is RS, 2 would be, PQ are 1 1 they remain as 1 1 here and here so they have to remain in the expression, here S is 1, here also S is 1 where R is 0 here R is 1 here so R changes from 0 to 1 between these two cells so R gets knocked off. R gets knocked off from this cell to this cell whereas S remains as 1 so 2 is nothing but PQS. For 3 it is the same argument PQ R remains 1 S changes from 1 to 0 S gets knocked off R remains so 3 is PQR.

Finally the fourth one is PQR 0 0, R is 0 here R is 1 here R is 0 here R gets knocked off here S remains 0 so it is P bar Q bar S bar. That means my final expression is this RS plus PQS plus PQR plus P bar Q bar S bar. Each of these terms 1 2 3 4 which are product terms which combine to become sum of product later on is called a prime implicant.

(Refer Slide Time 47:54)



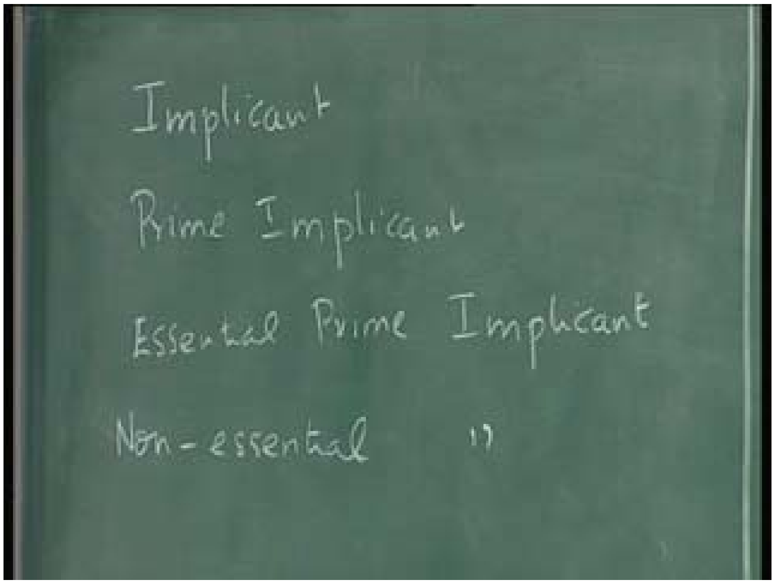
An implicant is any group of 1s. Suppose I write a term I won't do that, supposing I did between these two ones that means it is RS 1 1 P 1. So, if I write P RS P RS will mean these two cells cell number 11 and 15. Cell number 11 and 15 if I combine together I will write it as PRS. I will not do it because I am covering a larger group of 1s four 1s so there is no need to do this. But suppose I did it it's called an implicant, it's not a prime implicant.

A prime implicant is a largest possible group of 1s in that particular context for that group, you cannot find a larger group. An implicant gets submerged into a prime implicant. A prime implicant is a largest possible group of 1s that you can find for a given group out of all the prime implicants.

Any prime implicant which is essential for the final representation, I told you two things you have to make sure all 1s should be covered irrespective of the fact that 1s can be covered more than once. For example, this one has been covered 3 times this was covered along with this, this was covered along with this (Refer Slide Time: 47:15) so that is okay. But in each time we have included at least one 1 at least one cell which has not been combined earlier. When I wrote this number 2 term this one has been included new, when I wrote three terms I included this one new.

I defined an implicant, I defined a prime implicant and now I am defining an essential prime implicant. In an essential prime implicant there will be at least one 1 or one cell which has not been covered in any other prime implicant. there should at least be one cell or one true value which has not been included in any other group so such a prime implicant is called essential prime implicant in this case all the four are essential prime implicants because this one and this one cannot be covered and these three 1s cannot be covered other than this group, this one will not be covered other than this, this one will not be covered other than, these two 1s will not be covered in any other way. So here all the four prime implicants are essential prime implicants. Sometimes occasionally we will get non essential prime implicant.

(Refer Slide Time 49:27)



Supposing you have different ways of combining a 1 in two groups you can combine this way or that way both of them will lead to the same result or similar results. Such a group is called non essential prime implicant. That means there is a more than one way of combining a 1 which is not covered otherwise as a essential prime implicant. But one of them is essential otherwise that 1 will be left out.

Suppose I have two different ways of combining a particular 1 I have exhausted all 1s in a map, all entries in a cell for which the output is true I have exhausted except 1 there is a 1 hanging out which I have not covered yet in any essential prime implicant or any prime implicant. Now I want to combine this 1 and I can find several possibilities or at least two possibilities but one of them may be required the other may not be, if I take this the other one is not required if I take the other one this is not required. So this is a non essential prime implicant. prime implicant is important it has to be covered but each of those two prime implicant which I will get by combining this one with two different ways are called non essential prime implicants so it is not necessary to include all non essential prime implicants for the final output but it is only necessary to include as many non essential prime implicants as required to cover all the ones in the map. Whereas essential prime implicants all of them should be covered in your final output. We will take an example of non essential prime implicant, how a particular one can be covered in different ways and how only one way may be required and all other ways may be redundant that will tell you more about a non essential prime implicant. We will see it in the next lecture.