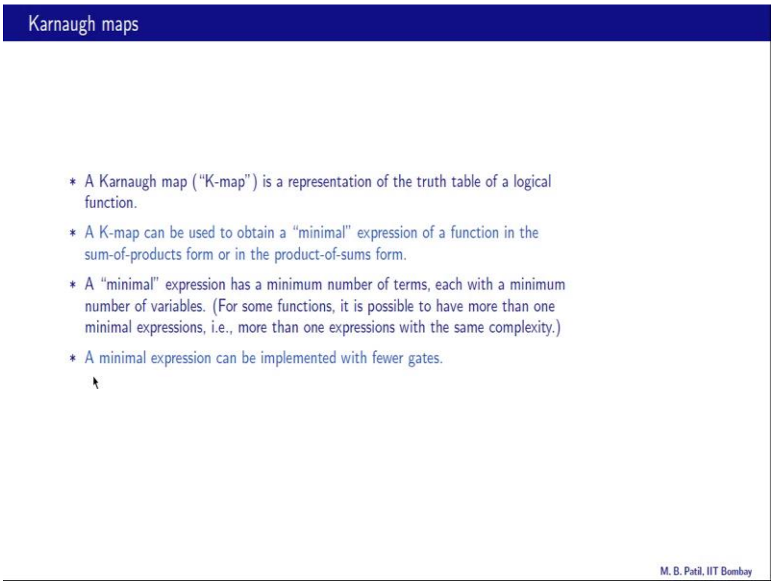
Basic Electronics
Prof. Mahesh Patil
Department of Electrical Engineering
Indian Institute of Technology, Bombay

Lecture – 57
Karnaugh maps

Welcome back to Basic Electronics. In this class, we look at the use of Karnaugh maps or K-maps in short to minimize a given logical function. First, we will see how a K-map is constructed; we will then look at how to write the minimal form of a logical function from the associated K-map. Finally, we will take up some examples in which the truth table of the logical function has do not care values. Let us get started.

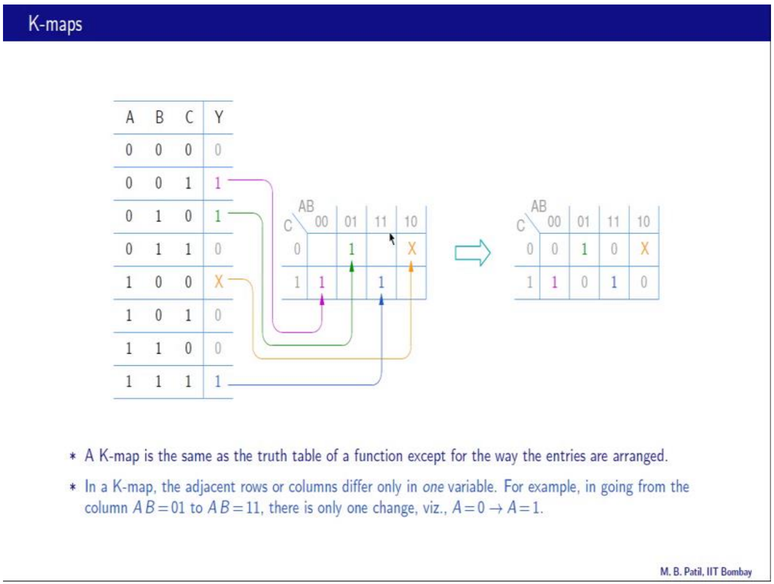
(Refer Slide Time: 00:50)



We will now discuss minimization of logical functions and good way to do that is to use Karnaugh map or simply called the K-map. What is the K-map? It is a representation of the truth table of a logical function. And we will see several examples of K-map. A K-map can be used to obtain a minimal expression of a function in the sum of products form or in the product of sums form. Now, in our course, we will mostly deal with the first 1 the sum of products form, but once you understand how that is happening, you should be able to pick up very easily the product of sums K-map as well.

The question that we should ask ourselves is; what is a minimal expression. Here is the answer: minimal expression as a minimum number of terms and each term has a minimum number of variables, so that is what minimal means. The next question that arises is whether each logical function has a unique minimal expression or not. The answer to that is mostly yes, but for some functions, it is possible to have more than one minimal expression. Now, why are we interested at all in minimal expressions this is the reason. A minimal expression can be implemented with fewer gates and that is why the implementation is cheaper.

(Refer Slide Time: 02:36)

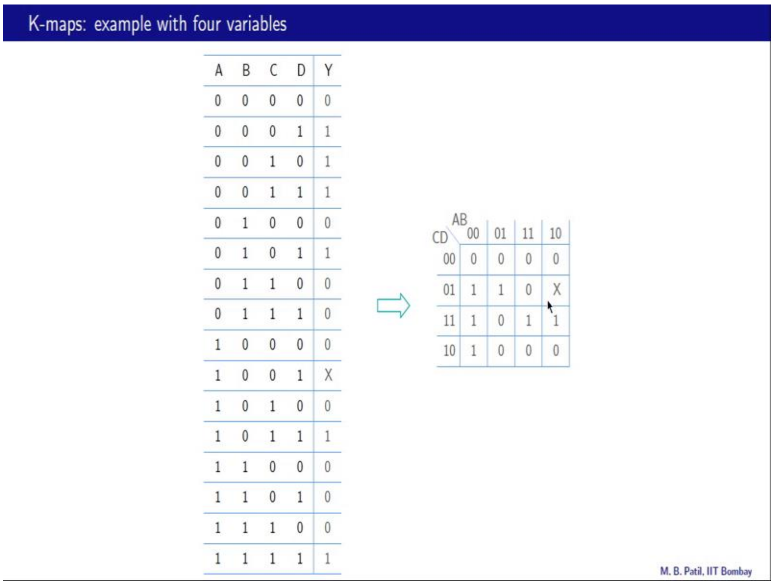


So, let us look at an example of K-map now here is a logical function Y of three variables A B and C we have 1's here 1 1 1. And we have one do not care condition here, and the rest of the entries are 0. So, let us begin by drawing this map, and let us see what this means. This is C, and this means C is 0 here, C is 1 here. These correspond to the values of A and B; here A is 0, B is 0, A is 0, B is 1, A is 1, B is 1, A is 1, B is 0. The next step is to map the logical function Y from this table to this map.

So, let us do that. We will start with this one here. What are A, B for this entry 0 0, so that is this column here. What is C that is 1, so it is this row here? So, we are talking about this box. So, this one will go into that box there like that. What about this one A B are 0 1, so this column and C is 0, so this row. So, we are talking about this box like that. What about this X, A B are 1, 0 this column, C is 0, so this row. So, we are talking about this box. And this last one here A B are 1 1 and C is 1, so this column and this row. So, these boxes here like that. Now, what about the boxes which are not filled so far, those will get these 0s; and with that we get our complete K-map shown here.

Let us now make a few observations. One we observed that a K-map is the same as the truth table it has the same information, except for the way the entries are arranged. Second and this is a very important point, if you do not remember this point then the results can be disastrous. So, therefore, we need to pay attention to this very carefully. In a K-map, the adjacent rows or columns differ only in one variable. For example, in going from the column A B equal to 0 1 - this one to a be equal to 1 1, there is only one change and that change is in variable A. Similarly, when we go from 1 1 to 1 0, there is a change in B only one change and so on.

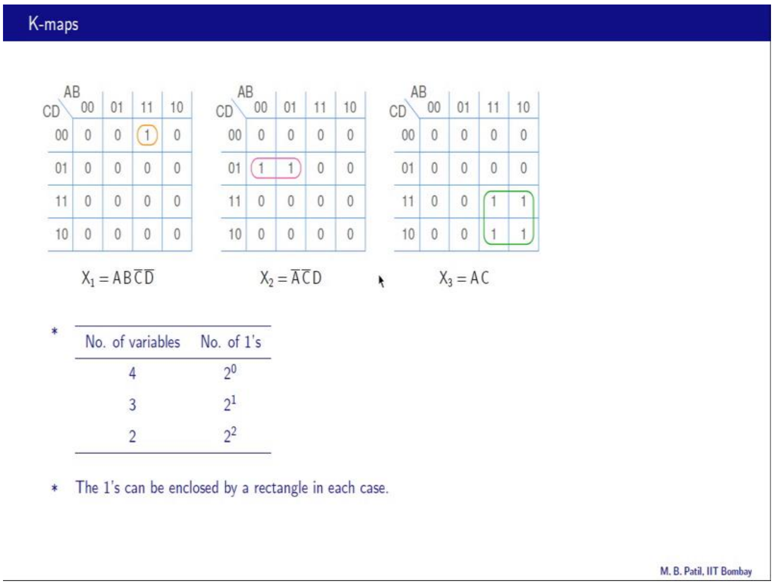
(Refer Slide Time: 05:55)



Here is a K-map with four variables. So, Y is a function of A, B, C and D. And here is the associated K-map. Let us see how this works. These values 0 0 0 1 1 1 etcetera correspond to A and B, these values correspond to C and D. And as we go from one column to the next, we notice as we mentioned earlier that there is only one change for example, from 0 0 to 0 1, B changes, but not A from 0 1 to 1 1, A changes, but not B and so on.

Similarly, as we go from one row to the next there is only one change going from 0 0 to 0 1, D changes from 0 to 1, but not C etcetera. Let us now look at the mapping of Y from this format to this format for a few entries. For example, consider this one. What are A and B - 0 0, what are C and D - 0 1. So, we have A B - 0 0 and C D - 0 1. So, we have this column and this row and that is where that one goes. Let us take this X for example, we have A B equal to 1 0, and we have C D equal to 0 1. So, A B equal to 1 0 is this column, and C D equal to 0 1 is this row and that is why we have that X over there etcetera. You can check the other entries as well.

(Refer Slide Time: 07:44)



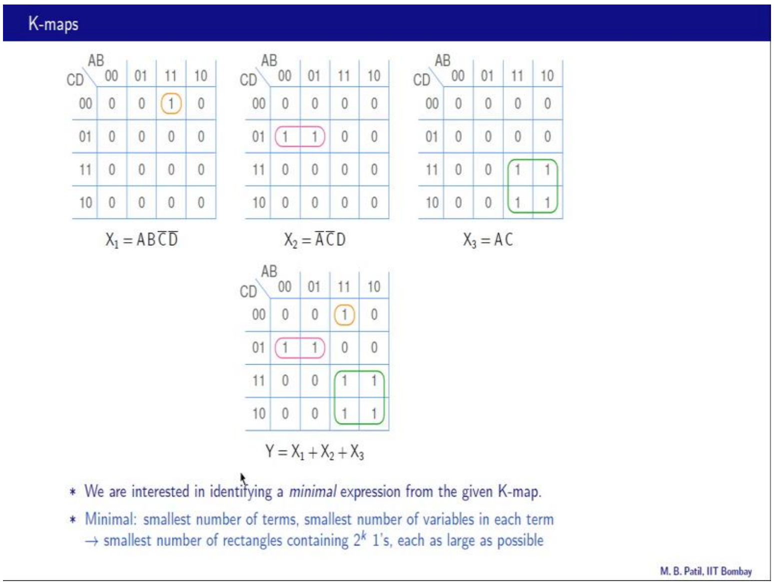
Let us now look at a few examples of functions with one single term such as this one here. So, this is a single product term, it is the AND operation of A, B, C bar and D bar. And when is this equal to 1 is 1 for A B equal to 1 1 and C D equal to 0 0, so that column and that row, that is where this one will appear. Let us consider another product term X 2, now this term has only three variables A, C and D. And X 2 is 1, if A is 0; that means, two columns because A is 0 here, A is also 0 here, so these are the two columns. And what is C D, C D would be 0 1 that is this row. So, therefore, we have two 1's now like that. Let us take another example X 3 equal to A and C this is another product term, a single product term. When is X 3 equal to 1? When A is 1 and C is 1; A is 1 in these two columns and C is 1 in these two rows, and therefore we have four 1's here.

Now, let us make few observations which are related to functions of this type, where the function is a single product term, but with different number of variables. Here, we have four variables here, we have three variables, and here we have two variables. Here is a table which relates the number of variables in these functions and the number of 1's which appear in the K-map. If we have four variables as in this case, then we have 1 1 there. If we have three variables like in this case, then we have 2 raise to 1 1's that is two 1's like that.

And if there are two variables as in this case, then we have 2 raise to 2 or 4 1's in the K-map like that. So, in each of these cases, the number of 1's is given by a power of two and that is a very important point to remember. So, that was about the number of 1's and now we want to talk about the position of these 1's. How do these 1's appear in the K-maps? And if we look at these examples we realize that the 1's can be enclosed by a rectangle in each case. So, here is A rectangle, here is a rectangle and this of course, is just a single entry.

Now, these 1's appear adjacent to each other and not 1 here and the other one here, so that we can form a rectangle around these two entities and the same thing holds about these four 1's as well, so that is a very important point as well.

(Refer Slide Time: 11:24)



Let us look at this; same examples once again and summarize the points that we made in the last slide. If we have a function which is a single product term as in all of these then we can make two points about the K-map. Point number 1, the number of 1's is given by A power of 2 - 2 raise to 0, 2 raise to 1, 2 raise to 2. And point number 2, the 1's is positioned such that we can draw a rectangle around them. Now, with these in mind let us proceed further.

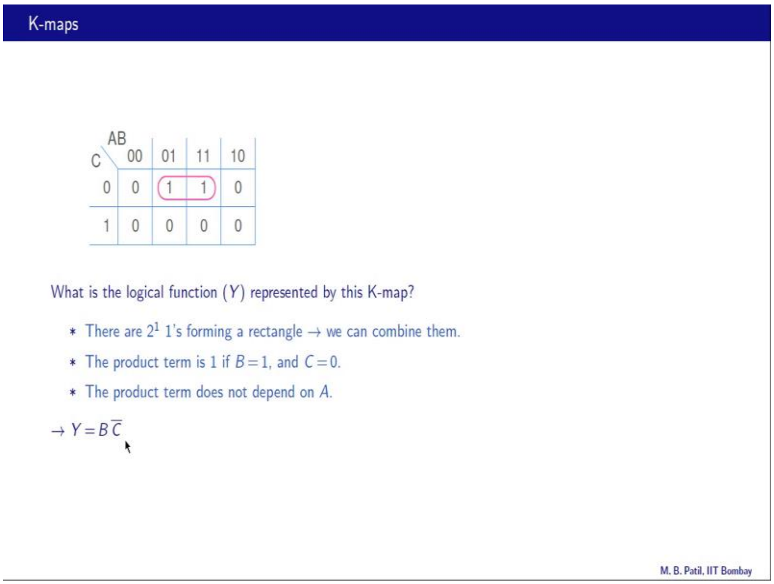
Let us now consider a function Y, which is the sum of X 1, X 2 and X 3 that is the OR operation of X 1, X 2 and X 3. What will the map of Y look like Y is 1 if X 1 is 1 or X two is 1 or X 3 is 1. So, all we need to do now is to look at the position of 1's in these three maps and just replicate those 1's in the map for Y, it is as simple as that. For example, this one, we will go here these two 1's will go here, and these four 1's will go over here and that gives us the map for the function Y.

We now come to the actual problem that we are interested in. We are interested in identifying minimal expression from the given K-map. So, take this Y as an example, we are given this map. And from this map, we want to find the minimal expression for the function. And what do we mean by minimal, we must have the smallest number of terms and smallest number of variables in each term that means, we must identify the smallest number of rectangles containing 2 raise to k 1's each as large as possible.

Let us see what this means in the context of this particular example. So, this K-map is given to us and we are asked to find the corresponding minimal expression. How do we go about it, we first identify rectangles which contain only 1's and no 0s such that the number of 1's is A power of 2. So, we identify rectangles with two 1's or four 1's or eight 1's and so on. Now we know that for each rectangle we can write a product term we have not yet looked at how to do that. So, there are three terms because we have three rectangles here this is of course, a rectangle containing a single term, so one rectangle, second rectangle third rectangle.

So, we know from this map that we have three terms and then we proceed to write each of these terms and that gives us the minimal expression. So, let us do that for a few examples and then it will become very clear.

(Refer Slide Time: 14:51)

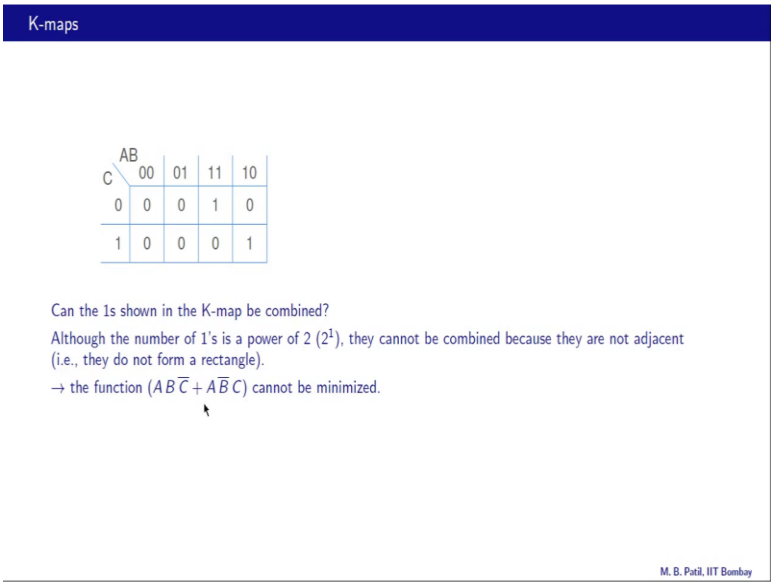


So, here is our first example, and we want to find the minimal expression for this logical function Y given by this K-map. So, what is the first question, we asked we want to find out how many 1’s are there; and we see that there are two 1’s. And in this case there happened to be adjacent, so we can have one rectangle covering these two 1’s like that. And since there are w raise to 1, 1’s forming a rectangle we can combine them. Since the rectangle covers all the 1’s which are in the K-map - these two 1’s, we can conclude that the logical function Y is represented by a single product term.

The next question is what is the product term? Let us make A few observations. One - the product term is 1 if B is equal to 1 and C is equal to 0. And we can see that from the K-map these are the two 1’s which are covered by our product term; for this 1 the value of B is 1; for this 1 also the value of B is 1. What about C in both these cases, C is 0 and that is why we say that the product term is 1, if B is equal to 1 and C is equal to 0 that is our first observation. Second - the product term does not depend on A; and we can see that from here; the product term is 1, if A is 0; it is also 1, if A is 1. So, clearly it does not depend on A.

So, the only product term which satisfies all these conditions is B and C bar. This product term is 1 if B is 1 and C is 0, and it does not depend on A. And since in this case, this product term represents the logical function Y, we can say that Y is equal to B C bar. And we can see that from here, the product term is 1 if A is 0; it is also 1, if A is 1. So, clearly it does not depend on A. Now, it turns out that the only product term that satisfies all of these conditions is B C bar; B C bar is 1 if B is 1 and C is 0; and B C bar does not depend on A. And in this specific example, the product term that we have figured out is the logical function Y itself, because as we mentioned earlier Y has a single product term because this one single rectangle covers all the 1's in the K-map, and therefore we conclude that Y is equal to B C bar.

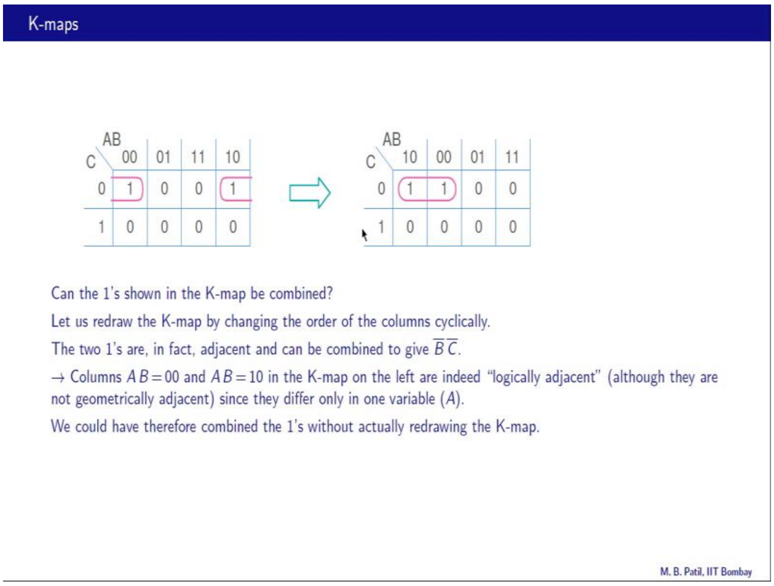
(Refer Slide Time: 18:17)



Next example, once again we have two 1's here, this one and this one. And the question is whether we can combine these two, the answer is although the number of 1's is a power of 2, they cannot be combined because they are not adjacent that is they do not form a rectangle. There only rectangle we can think of which will contain these two 1's is this one, but then in that case it will also contain 0s and that is of course, something we do not want. So, then we have a function with two terms which cannot be combined and each of them is a minterm.

What is this minterm? It is 1 when A is 1, B is 1 and C is 0, so it is A B C bar. What is this minterm, it is 1, when A is 1, B is 0, and C is 1, so it would be A B bar c. So, our function then is this term plus this term like that A B C bar plus A B bar C, and it cannot be minimized further.

(Refer Slide Time: 19:37)

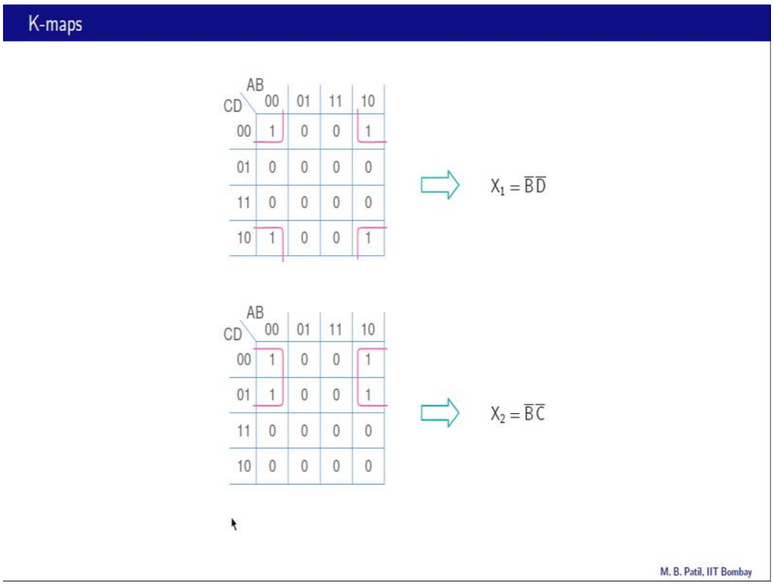


Next example, once again we have two 1's, one here and one here. And the question is whether these two 1's can be combined. Now, it seems to us that we cannot draw 1 tangle without also including these 0s to contain these two 1's that we have and so the answer seems to be that the 1's cannot be combined. But let us redraw this K-map by changing the order of the columns in a cyclical fashion, and see what we get. Here is the revised K-map and what we have done here is we have drawn this 1 0 column first followed by 0 0 then 0 1 and then 1 1, so 0 0 0 1 and 1 1. And these two entries 1 0 have gone here these two entries have gone here and so on. So, by doing that we are not really changing their function, the function is still what it was, all we have done is rearranged the entries in the K-map. And now we see that this one's have come next to each other.

So, the two 1's are in fact adjacent and can be combined like that. And what is the term that corresponds to this rectangle that is easy to figure out. We know that that product term is 1 if B is 0 and if C is 0, and it is independent of A, because A is 1 here, and A 0 here. So, that term must be B bar C bar. In other words, columns A B equal to 0 0, and A B equal to 1 0 in the K-map on the left are indeed logically adjacent. So, these two columns 0 0 and 1 0 are logically adjacent. And what does it mean; that means, if we go from this column to this column, there is a change only in one variable.

So, A is changing from 0 to 1, but B is not changing. So, therefore, these are logically adjacent although they are not geometrically adjacent. And we could therefore have combined these 1's without actually redrawing the K-map like we did here. As long as we know that these two columns are adjacent logically adjacent, we can think of a rectangle, which covers this 1 and this 1.

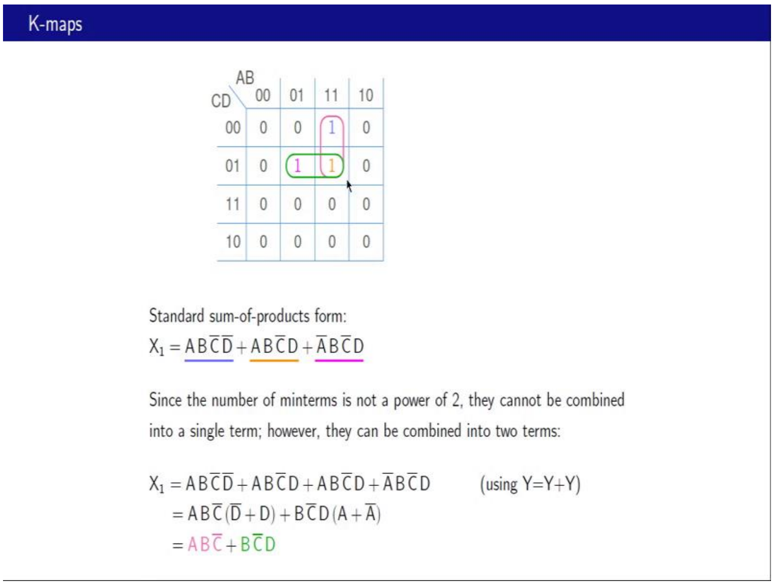
(Refer Slide Time: 22:24)



Let us take this example. How many 1's do we have 1, 2, 3 and 4. And it appears that these four 1's are not adjacent to each other, and therefore, cannot be combined. But they actually are adjacent, and why is that because this column is adjacent to this column logically adjacent. Similarly, this row is adjacent to this row, and therefore it is actually possible to draw a rectangle, which will cover all of these four 1's like that. And you are definitely encouraged to redraw this K-map, so that these four 1's will appear as a bunch. So, what is the term that corresponds to these four 1's that product term is independent of A, because A is 0 here and A is 1 here, it has B bar because B is 0 here and B is 0 here. It is independent of C, because C is 0 here, C is 1 here; and it contains D bar, because D is 0 here, D is 0 here. So, that product that we are looking for is X equal to B bar D bar.

Another example once again we have four 1's – 1, 2, 3, 4. And we can think of a rectangle which can cover these two 1's, another rectangle which can cover these two 1's. So, our function can be written as A sum of two terms 1 corresponding to this rectangle here and another corresponding to this rectangle here, but we can actually do better than that because this column 0 0 is adjacent to this column 1 0. And In fact, these four 1's can be combined with a single rectangle like that. And what is the term that corresponds to this rectangle, it must be independent of A, because A 0 here, 1 here, it must contain B bar because B is 0 here. It must be independent of D, because D is 0 here 1 here; and it must contain C bar because C is 0 here, and that gives us B bar C bar.

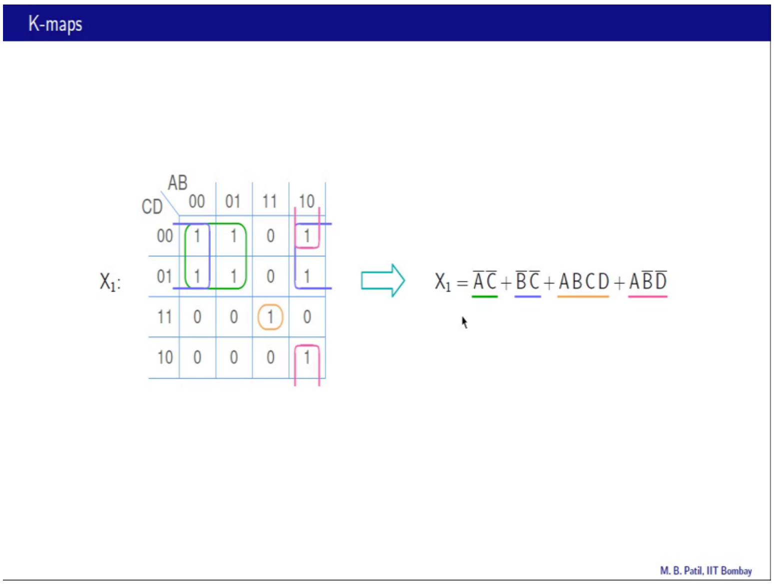
(Refer Slide Time: 25:05)



Let us now consider this example, which brings out a very important point. We have three 1's here. This 1 corresponds to A B C bar D bar, this first term here. This 1 corresponds to A B C bar D - the second term. And the third 1 to A bar B C bar D - the third term here. Now, since we have only three 1's and 3 is not the power of 2, these 1's cannot be combined into a single term; however, they can be combined into two terms and let us see how. We can in fact, have two rectangles one like that and one like that which will cover these three 1's, but then notice that this 1 is getting covered twice, now is that ok?

Let us check let us write X 1 as A B C bar D bar plus A B C bar D. And we will write this term twice, so A B C bar D plus A B C bar D and then this term. Now, these two terms have A B C bar common and then we get D plus D bar like that; and these two terms have B C bar D common, and then we get A plus A bar. Now, this is 1, this is also 1 and that gives us this minimal expression A B C bar plus B C bar D. And we do not need to actually go through this algebra. We can simply do this by inspection. We can draw two rectangles 1 to cover these two 1's and another to cover these two 1's. And it appears that we have covering this 1 two times, but that is and that is because we have use this identity here. This term call it Y, we have written as Y equal to Y plus Y and that is perfectly fine.

(Refer Slide Time: 27:37)

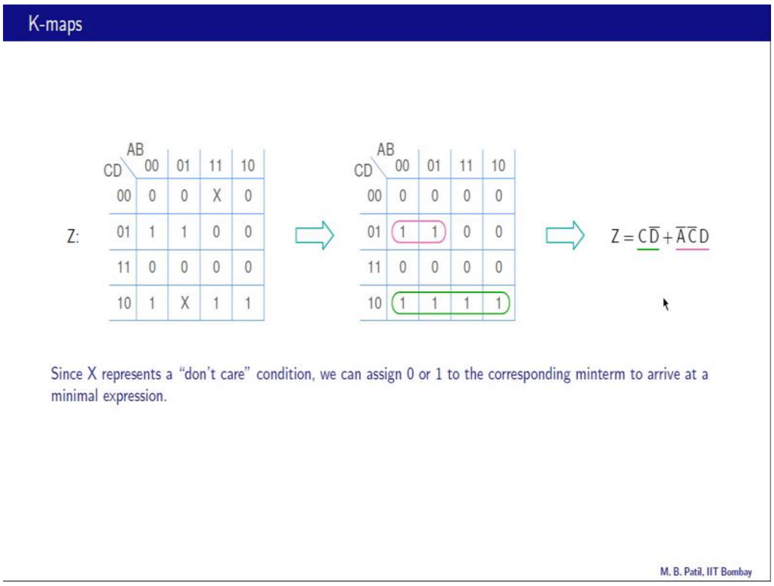


One more example, these four 1's here can be clubbed into a single term; these two 1's can be combined with these two 1's here, because this column and this column are logically adjacent. What will happen if we combine only these two, and not these four? This rectangle would have only two 1's and that will mean a product term with three variables. Whereas, this rectangle the purple one has four 1's in it and therefore, that gives us a product term with two variables. So, we save one variable in going from two 1's to four 1's. Now, this 1 cannot be combined with any neighbor and therefore that remains as a minterm; this 1 is adjacent to this 1, because this row and this row are logically adjacent, and therefore we can combine that 1 with this 1.

Notice that some of the 1's have been used multiple times. And like we saw earlier, we can do that because of the identity Y is equal to Y plus Y. For example, this 1 has been used in the pink rectangle as well as in the purple rectangle. This 1 has been used in the purple rectangle as well as in the green rectangle. We can use 1's multiple times and what is the overall objective of this whole exercise, we want to cover all the 1's in the key map with the smallest number of rectangles that gives us the smallest number of product terms in the final minimal expression, and with each rectangle as large as possible. And what does that do for us that gives us product terms with a smaller number of variables, so that is how we get minimal expression.

Now, let us see what the minimal expression corresponding to these rectangles is there. This term for example, A bar C bar corresponds to this green rectangle here, and you should really verify that. So, we have four terms in the minimal expression one corresponding to the green rectangle, one corresponding to this purple rectangle this one, one corresponding to this yellow rectangle which is just a single min term and one corresponding to this red rectangle that one, so that is our final minimal expression.

(Refer Slide Time: 30:43)



Let us look at this example which has some do not care conditions. There is a do not care condition here denoted by X, and there is another one here. Now, the question is what do we do with these X's. We can make this X, 0 or 1, similarly we can make this X, 0 or 1 and the question is what would be A good choice let us see. Suppose, we make this X equal to 0, and this X also equal to 0, what do we have then, then we have this 1 here this 1 here this 1 this 1 and this 1 we have five 1's these two 1's can be combined with a rectangle these two 1's can also be combined with a rectangle. This 1 is actually adjacent to this 1 here, and therefore these two can be combined as well. So, we have three rectangles that means, three product terms in the final expression, and each term will have three variables, so that is the situation and it turns out that we can do better than that and let us see how.

Let us suppose that we make this X equal to 0, and make this X equal to 1. And now we find that we have a much better situation these two 1's can be combined with one rectangle and these four 1's can be combined with another rectangle. And with that we have only two terms in the final expression C D bar plus A bar C bar D and that definitely is a better choice than our first choice in which both X's were 0s. So, when we have do not care conditions in a K-map we need to consider various possibilities and choose that possibility which gives us the minimal expression.

In summary, we have learned how to use K-maps to minimize logical functions. We will find this technique very useful in subsequent topics that is all for now. See you next time.