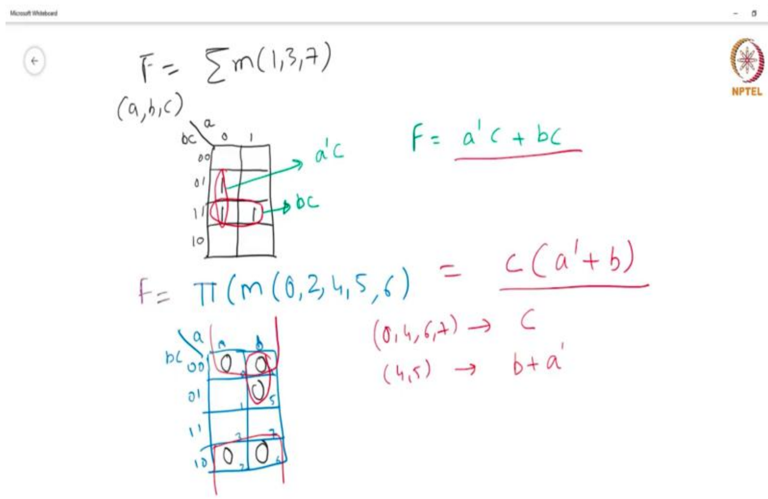
Digital System Design

Professor Neeraj Goel
Department of Computer Science Engineering,
Indian Institute of Technology Ropar

Boolean Minimization Using K-Maps

(Refer Slide Time: 00:14)



So let us take couple of examples for doing Karnaugh map minimization. So let us take our first example. Now let us say F is the represented using sum of minterm 1, 3 and 7. So because it has going to be like we say that this is this will have this 3 variables. And let us try to follow one terminology that the variable which is written at 1st place is the most significant variable and the variable which is written in the end is the least significant variable.

So that would help us in having one to one correspondence with these minterms. So how to create a K map for this? We generally write a here, we write b and c and there has to be these 8 cells. So this cell we can say represent 0, this cell represent 1, this is 00 01 11 and 10. So because minterm 1 would mean that 00. So this has to be 001 so that means this has to be 1. Now for minterm 3, a has to be 0 and b and c both are 1.

So we can write 1 here and for minterm 7 we can say abc, a is 1, b and c both are 1. Now this is how we will fill this minterm truth table and this K map. Now we can find out what are the adjacent terms which can be minimized. So you can see this 1 and this 1 form the adjacent term. This is called 1 cube. So similarly this 1 and this 1 can be also combined and form another cube.

This 1 and this 1 cannot be combined because they are diagonal, they are not adjacent. So each of the combined group we can then say that what would be the minimized expression. So for example 4 this 1, the minimized expression would be 11 because this 0 and 1 will get minimized. So this I can write as b and c. Similarly, this one, the thing which is 0 and 1 will get minimized.

So otherwise 0 is seen for both of it and c equal to 1 is also same. So b will get minimized. So I can write this as a dash and c. The whole function F can be written as a dash c plus bc. So this is how we can come up with the minimized expression. So similar to this you can also, let us do it for the max term expression. Let us say the same function F can be written as the same function F can also be written as the max term.

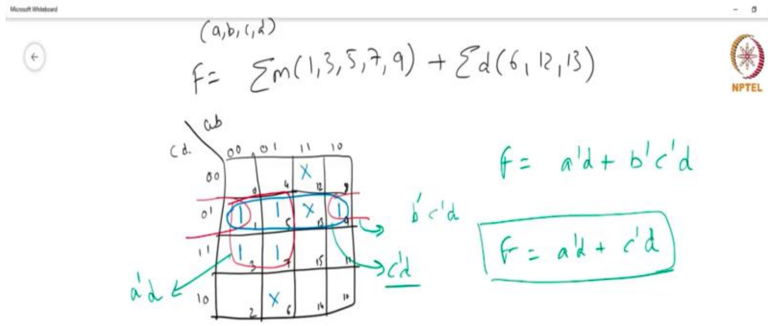
So let us say for max term I will use this blue color. So this violet color. So this F can be written as, so this is product of m 0, 2, 4, 5 and 6. So again, I will draw the, draw this table or a K map, so in the similar way this would be a here. This is b and c, this is 0, this is 0, this is 00 01 11 10. So the another way which help us sometime is we write in this, in the in one corner that this is 0, this is 1, this is 2, this is 3, this is 4, 5, 6 and 7.

Now for this term these are the 0. So I will put them in let us say in the black so that it is more visible, will make it in the black. So this is 0 and 2 is and 4, 5 and 6. So what all can be combined? So here the combination because these 2 can be combined and further these 2 and these 2. They also form they form a square so we can combine them and these 2. So essentially these something we also write these groups.

So this can be there are 4 groups. So one is 0 4, 6 and 7, this is one group. The other group is 4 and 5. So corresponding to this group, what would be the term? The term would be because this was 0, this is 1. Yeah, so this was 1 so 0 and 1 this that means a term will not be there. Similarly, b is 0 and 1 here, so b will not be here, so only c would be there. So corresponding to this the term would be and c is 0 so it has to be c.

And 4 and 5 this will become, so this is 1. So that means it has to be a dash and for there will not be c term because c is 0 and 1 here. So c because of this pair c will go away. So this is b plus a dash so I can write the whole expression also as product c, a dash plus b. So if you see that this expression and this expression are actually equivalent. So this is how we can create using a sum of product or using product of sum. So let us try out one more example.

(Refer Slide Time: 07:32)



Let us say our, we wanted to minimize, let us say we want to do minimize something with 4 variables. So with 4 variables and let us say it also have these don't care conditions. So we use something which will have don't care conditions also. Let us say this F is sum of minterms 1, 3, 5, 7 and 9. And let us say it also have some of the don't care conditions which are 6, 12 and 13.

So we will try to do it without don't care conditions. And we will also do it with we don't care conditions. So first of all, let us quickly draw our K map. So while drawing the K map because it is a 4 variable so we will create 16 cells and let us say a and b are here and c and d is here. So this function is of let us say a, b, c, d. So that means a form the highest significant bit and d form the least significant bit.

So we write the gray code here like 00 01 11 10. Here also we write 00 01 11 10. And in the corner of each cell, we write their corresponding minterm. So that it is easy for us to remember. So let us say we are writing it 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14 and 15. So after writing the minterms and creating this map then we will fill up the terms. So this term is 1, 3 is 1, 5 is 1, 7 is 1 and 9 is 1.

So don't care conditions instead of writing 1 and 0 we represent them using X. So 6 is X and 12 and 13 are also X, where is 13? 13 is here and 12 is here. So now I want you to create cubes or I wanted to combine the terms. So we will see that we can see a combination here which can form a queue. So like this two, 1 and 3 were adjacent, 5 and 7 were adjacent, 1, 3 and 5, 7 were adjacent so we created a bigger cube.

The other possibility was that we could have done this and this so we can make a group of 1 and 9. So corresponding to 1 and 9, the variable would have been so the min, the product term would have been. So we see what changes, what does not changes, here 0 and 1 does not change for this row. So we can directly write c dash d and for this term and this term a is 0 and 1 so a can be minimized.

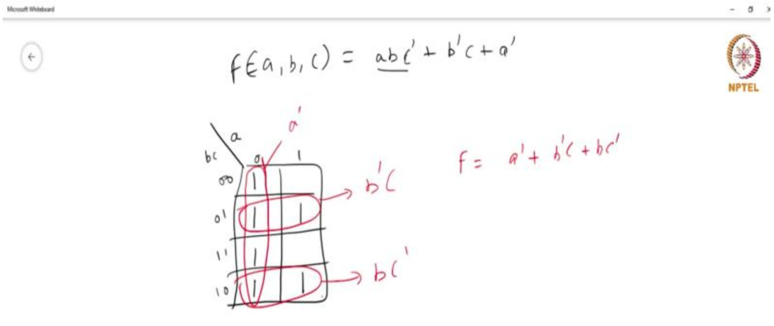
But b is seen for both of these minterms. So we can write b here. b is, would be there in the complemented form. So this is b dash. So what about this? For this you see that c would be optimized because c is 0 here and 1 here but d will, d is constant so d will remain. And similarly we see on the top, so we see that b would be optimized because b is 0 here and 1 here but a is constant for both of them so we can write a in the complemented form.

So if I do not consider don't cares, my function F would be written as a dash d plus b dash c dash d. So this is how we can, we can optimize. Now let us say this don't care conditions are also given. So don't care conditions means these input combinations will never come or will not be possible. So because these combinations will not be possible so what we can do is we can, if those conditions are enough favorable, favorable cases.

So in those favorable cases we can combine them to create something which would be of minimum form. So for example, here this particular X, this 13 can help me in creating a cube which is bigger. So I can go ahead and create this cube. So this cube will have less number of literals and this cube would only be c dash d. So the if I am including these don't care terms then my expression F can be written as a dash d plus and c dash d.

So this function can be represented which is this function F is a more optimized function because it has less number of literals although number of terms are same but literals in this case is lesser so this is, this forms the more optimized version. So this is how we can solve the K maps for 4 variables. So let us do one more thing here. So we have some 5 more minutes. So let us try to clear, yeah.

(Refer Slide Time: 14:14)



So let us say we are now there is some expression which is available to us in not in form of mean terms but in form of a un-optimized expression and we try, we wanted to optimize this expression. So let us say the function is given as a function is of a comma b comma c and is equal to abc dash plus b dash c plus a dash. Now we would like to optimize such kind of a function.

So if we have to optimize a function which is given to us, now how we can do it is we can also create a K map corresponding to this. Now you see that minterms are not given. So even though minterms are not given, so we can still create a K map which could be of could be helpful here, that is why we are not writing in the in small 0, 1, 2, 3, 4, 5, 6. But we can directly write from here.

So this is abc dash will be a is 1, b is 1 and c is 0. So that means this is this particular minterm. So this is a is 1, b is 1 and c is 0. So here this is b dash c, b dash b is 0 and c is 1 but there is no a, so that means it has to be this as well as this. Now let us see a dash, a dash means this is 1. This is 1. This is 1. Now if I want to group and combine so I can see that this a dash is anywhere there. Now I can also combine these two. I can also combine these two.

So this is a dash we know and this will become b dash c and this will become bc dash. So the whole expression, then this F can also be written as a dash plus b dash c plus bc dash. So similarly we can take more examples. So maybe I will share couple of tutorial questions to you and then you can solve those questions and we can discuss them in tutorial. Thank you very much.