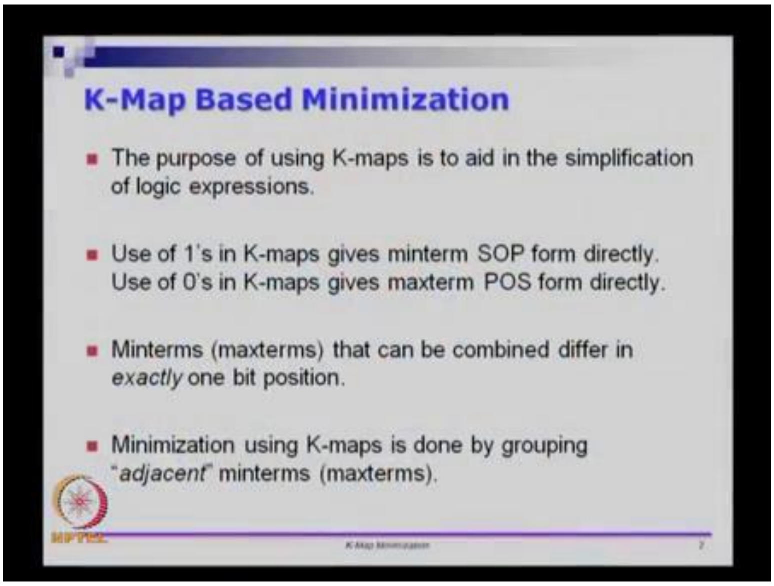
Digital Circuits and Systems
Prof. Shankar Balachandran
Department of Electrical Engineering
Indian Institute of Technology, Bombay
And
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Module - 9
K-Map minimization

Hello, we are at module nine. In this module what we will look at is, we will see how to use this so called Karnaugh maps or K-maps and how to do minimization of Boolean functions. Without using Boolean expressions explicitly, we will see how to derive simplified expressions for Boolean expressions.

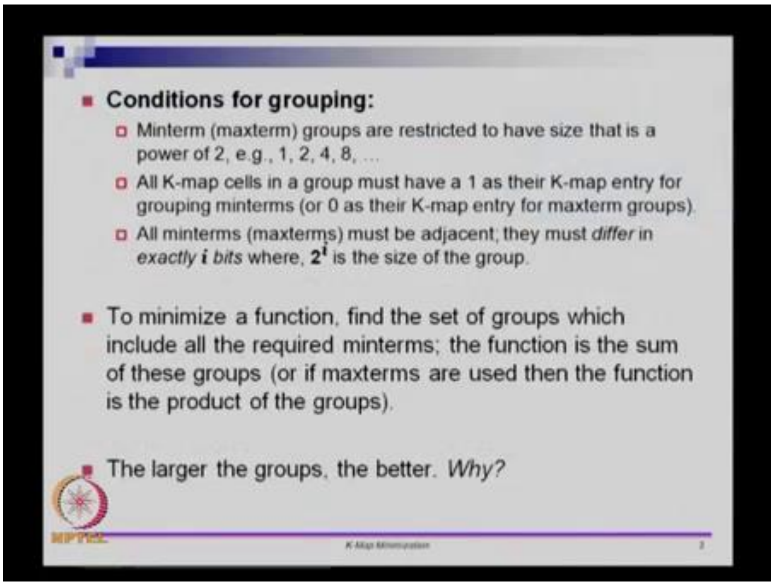
(Refer Slide Time: 00:39)



So, when you do K-map based minimization, the purpose of K-map itself is to give you visual way of doing thing. So, whenever we have 1 in the K-map, they correspond to the minterms in the sum of product form, and all the 0s correspond to the maxterms in the product of sum forms directly. So, this is something that is a, remember K-map is just a rearrangement truth table and wherever you have 1 in the truth table, you will have a 1 here; wherever you have a 0 in the truth table, you will have a corresponding thing, will have a 0 here.

Only that the way in which the table is arranged is going to give us something interesting. And we can combine minterms that are adjacent, adjacent to each other, which differ exactly in one position, we can combine them and minimization is done by using these adjacent terms and grouping term. So, we will see a detailed example in a little while.

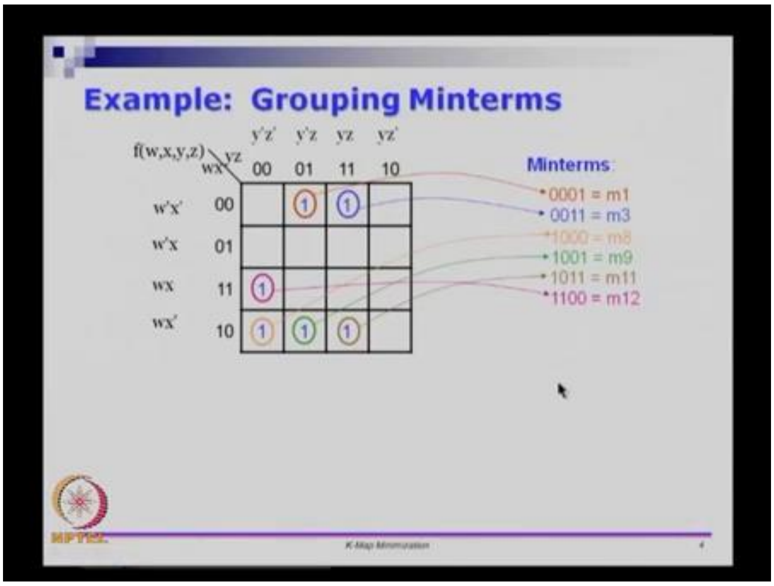
(Refer Slide Time: 01:30)



So, there are lots of conditions for grouping. The first thing is, when you have grouping, you are going to, so we are going to something called grouping, we are going to group the 1s together for some of products and we are going to group the 0s together for product of sums, but the grouping has to be done when you have powers of 2. So, you have to pick cells, the number of cells that you have should be a power of 2. Either you can have a group of one cell, which is just 1 or you can group two cells, which are adjacent to each other or 4 cells, which are adjacent to each other or 8 cells, which are adjacent to each other. You cannot do three cells and five cells and six cells and so on. So, it is 1, 2, 4 and 8 and what not.

And whenever you group, the next important criteria is, if you are going to group the 1s together, they must have all 1s if you trying to get minterms. And they must all have 0s if you are trying to get product of sums. If you are trying to get sum of products, the grouping that you are doing should all be on 1s and if you are going to do product of sums, they should all be on 0s and they must all be adjacent to each other. So, we will see an example and we will, I can come back to this slide little later.

(Refer Slide Time: 02:48)

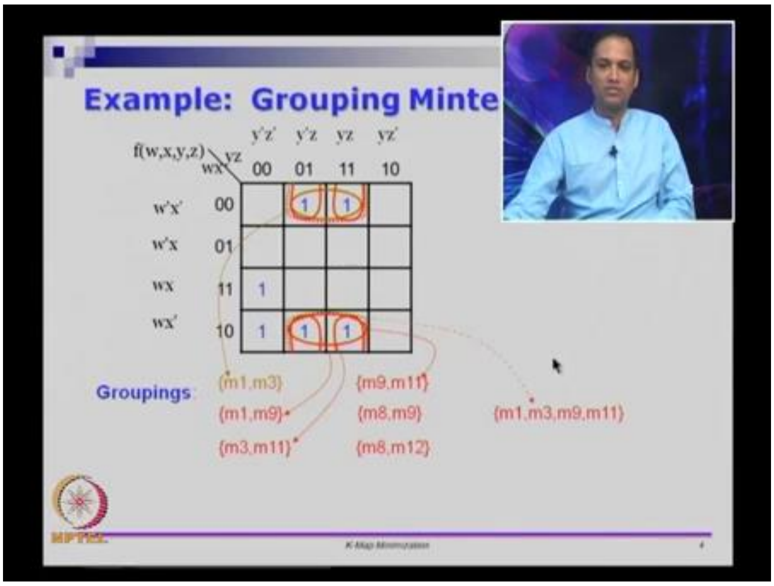


Let us see an example. Let us say I have, I am given a function like this. So, I have given a function in which six of the sixteen rows have 1s and this is the function on w x, y and z. So, what we have done is, we have laid it out in the form, that we have w bar x bar here, w bar x, wx and w x bar. They correspond to 00, 01, 11 and 10. Similarly, these four columns correspond to, yz being 00, 01, 11 and 10. Now, you go and look at this.

Let us, let us argue which of the 1s are adjacent and which of the 1s are not. So, if I look at the minterms, this is a single term. So, this is just grouping of just one cell. So, this is the grouping of one cell, and what is this cell giving us? It is for w bar x bar y bar and z or 0001. Similarly, this cell corresponds to w bar x bar y and z, right or this is m3. This term corresponds to w x bar y bar z bar. So, that is 1000 or m8. This corresponds to m9, this one corresponds to m11. Remember, it is 8, 9, jump to get 10 and comeback to get 11. So, this, this one correspond to m11. So, you have m8, m9 and m11. And this one corresponds to 1100 or m12.

So, here we have not really grouped multiple 1s together, but we have followed all the properties that we need for K-maps. We can only group powers of 2. In this case, each 1 is circle and each 1, it is grouped by itself. We have grouped 2 power 0 or 1 term together. So, each one is an individual term. This does not give you any minimization so far. If you took only these and directly wrote the expression, that would be sigma of 1, 3, 8, 9, 11 and 12. There is no minimization, all the minterms would be there.

(Refer Slide Time: 04:53)

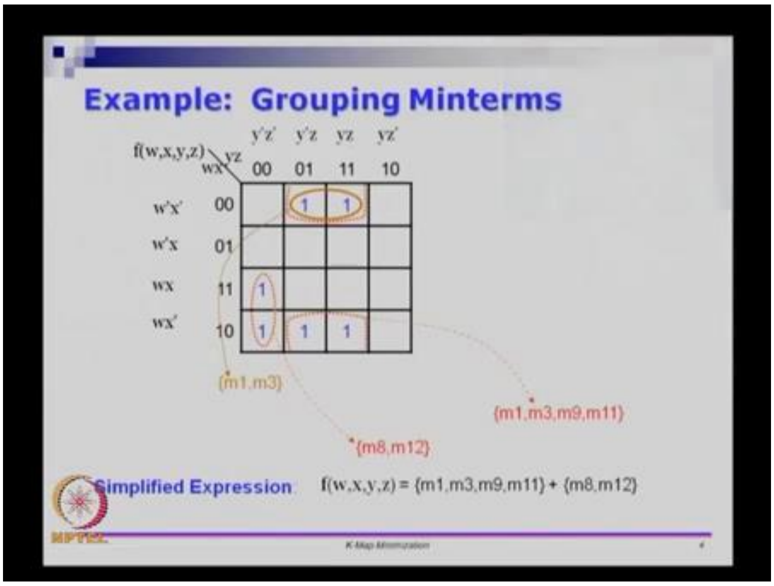


Now, let us look at grouping two terms at a time. You look at these two things, these two 1s are adjacent to each other because visually, they are adjacent to each other. So, this is the term corresponding to m1, this is the term corresponding to m3, right. m1 and m3 being adjacent to each other can be combined together. m1 and m3 differ only in the 3rd bit, namely for this y. So, y is 0 in this column and y is 1 in this column, right. So, when you combine them you get m 1 and m 3.

And remember, I said, you can role the paper out and so on. So, this and this can be combined together. So, m1 and m9 can be actually combined together to give term called m1, m9. These two can be combined together to give you term called m3, m11 because they are also adjacent to each other and finally, you can combine this and get m9, m11. You, there are several other combinations of 1s. So, for example, these two 1s can be grouped together, these two 1s can be grouped together and so on. So, all these are listed here, right. So, there are six possible combinations with which we can have two of them combined at a time.

So, now, let us take this last step. Can we combine four at a time? So, we have these two cells, which are adjacent to each other and that combination is adjacent to these two cells, which are also adjacent to each other. So, you have a combination of two, which is adjacent to another combination of two. So, you can combine these two and get a combination of four. So, if you go and look at this and this, that gives you the terms m1 m3 and m9 and m11. You can read it from the table also. This is m1, this is m3, this is m9 and m11. These four are combined together and that is a term also, right; that, that is one expression.

(Refer Slide Time: 06:52)

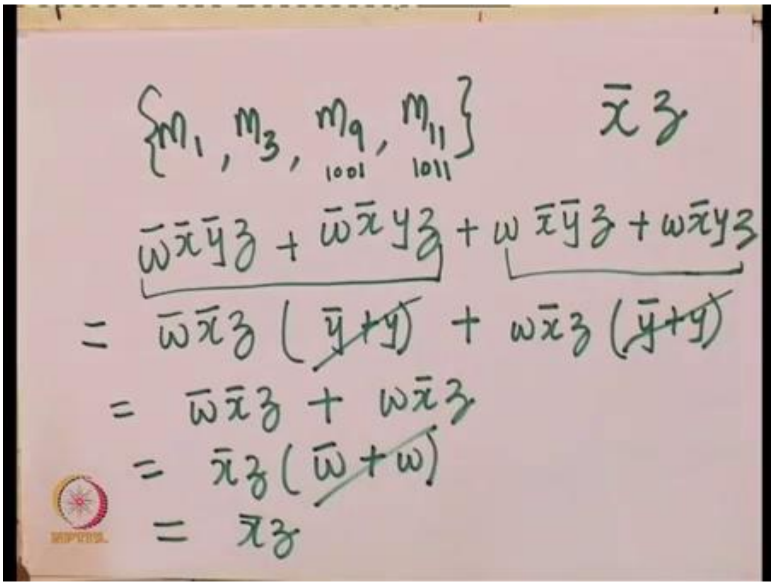


So, let us look at the simplified expression. So, we have a grouping of four here. So, forget this oval and the remark here, we do not need this, the minimize expression does not have it. So, we did several groupings. There are several grouping of 1, several grouping of 2 and several groupings of 4. We start with the largest group.

The largest group is grouping of four. These two are 1s and these two are 1s, the largest grouping is a group of four. I take that and that, correspond to the term m1 and m3, m9 m11, right, these four terms. Then, these 1s are already covered, however these two 1s are not covered by this group of four. We take that and the terms corresponding to that are m8 and m12, right; m8 and m12. So, this m1, m3 is, is this oval, but m1, m3 is a subset of m1, m3, m9, m11. So, m1, m3 need not be covered explicitly. So, we have m8, m12, which covers these two 1s; m1, m3, m9, m11, which covers these four 1s, we have all the 1s covered. So, which means, the expression is done, you can go and write f of w, x, y, z as sum of these two terms, this product m1, m3, m9, m11 and this m8, m12, right. We will see what it means to get this product m1, m3, m9, m11. Let us go and look at these two rows again.

So, this U kind of thing here and inverted U here that combines and gives you m1, m3, m9, m11. Let us see what it means. So, in these four 1s let us see what is common. So, w bar, x bar and w x bar. So, these are two different rows in which they are there. However, we see, that w bar is here and w is here, x bar is remaining constant, right. Across these two rows, x bar is remaining constant and across these two columns z is remaining constant. So, the intersection of that is m1, m3, m9 and m11 were x bar and z is what you need. So, this m1, m3, m9, m11 corresponds to the term x bar and z. So, this is something that you can possible try out. So, let us, let us do this. I will do on a piece of paper.

(Refer Slide Time: 09:22)



Let us look at the term m1, m3, m9, m11, right. So, I want to see how this gets simplified. So, I said, that this simplifies to x bar z, let us see why. So, m1 corresponds to w bar x bar y bar z. So, you can see, that 0001 m3 corresponds to w bar x bar y and z because it is 0011, 9 is 1001. So, that corresponds to the term w x bar y bar and z and 11 corresponds to 1011. So, I can write it as w x bar y and z. So, we have that.

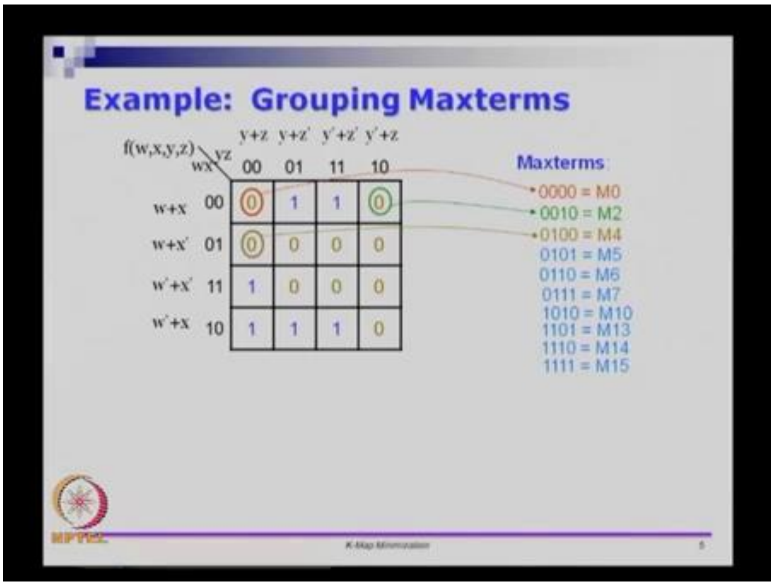
Now, let us see how this can be combined together. So, if I take these two terms what do you see common? Here, w bar x bar z is common, then you have y bar plus y. If I take these two terms here you can notice, that w x bar z is common aqnd you have y bar plus y do not you right. So, m nine and m eleven you have this we know this is one and this is one you can write it as w bar x bar z plus w x bar z. Again, here x bar is common to both. So, x bar z is common to both, we can write it as w bar plus w. This is again 1, this is the same as x bar z. So, you can do this all Boolean minimization and get x bar z. You can also get x bar z by just looking at the term in the table.

So, you look at this table here, in these four 1s what is something that is common? So, these four cells have x bar in the row, which contains x bar and in the column which contains z. However, y is changing from this column to this column and w changing from to this column to this column, those two terms will not appear in the product term that is what we get from m1, m3, m9, m11. So, think about what you will get from m8 and m12. Just for a moment think about what you get from m8 and m12, you can always go and write it as a Boolean expression and minimize it and so on. But if you want to do this visually, I will go to look at the m8 and m12. So, w is there which is not changing, x is changing. So, x will not appear in the product term. So, this is w y bar z bar that corresponds to m8 and m12. So, this terms corresponds to x bar z, this terms corresponds to m8 m12, which is w y bar z bar. So, this is the very nice and pictorial way of getting these things as opposed to doing all these complicated things that I showed on the paper. We can get to these product terms relatively quickly by knowing how to group.

So, let me go back and show you how the grouping is done, what are the rules of grouping let us see that. The conditions for grouping are, minterms groups are restricted to have size, that is, the power of 2, you can either group just the one terms or two terms together or four terms or eight terms together and so on. All the cells in the group must have 1 as their entry if you want to group minterms or if you want group maxterms, they can have 0s. You cannot mix 1s and 0s and all of them must be adjacent to each other. So, these are three different things that we want.

Now, to minimize a function you go and find out the set of the group, which includes all the required minterms. You should not leave any, any of the 1s uncovered. All the 1s should be covered using these ovals or rectangle or whatever you are drawing, right, you should cover them. The function is sum of only those groups if you do the sum of products. If you covered 0s, then you get the maxterms and the function is actually logical and of all those sums and in general, larger the group, it is better for you. Do not think about why, we will see this in a little while later, but you for now do not think about why having a larger group is better. So, if I have four 1s that are adjacent together, instead of keeping the four 1s separately, it is better to go and group all the 4s, all the four 1s together. Do not think about why it is so.

(Refer Slide Time: 14:16)



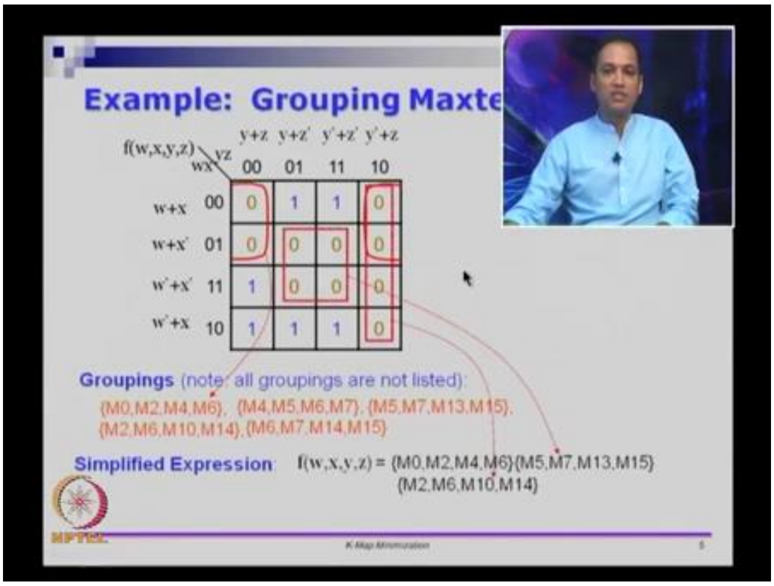
So, now let us see how the same thing can be done by grouping maxterms. So, for maxterms first thing we do is, we notice, that the 1s that we had earlier are all in blue and we have put all the other things as 0, which are in a different color, they are in brownish color, right. Now, the rows are maintained as, they are 00, 01, 11 and 10. 00, 01, 11 and 10. They correspond to wx and yz, but the interpretations of the rows are now going to be different. So, 00 corresponds to the sum term w plus x; 01 here corresponds to the sum term w plus x bar. This corresponds to w bar plus x bar and this corresponds to w bar plus x. So, remember, this is the logical complement of what we had earlier.

So, if you want to interpret it in this product form, you would have written it as w bar x bar, but if you want to interpret it in the sum form, it is w plus x. So, this would have been w bar and x bar. Now, you interpret it as w plus x. similarly, this column would have been y and z if you interpret it in the product form, but in the sum form, that is, y bar plus z bar. So, now, we have we are going to do this with product of sums. In the last example I did sum of products, in this example we will, I will show you how to do product of sums. For product of sums we are going to group the 0s together. So, we have several things, now let us see how to group them.

So, let us start with what are all different maxterms. Each 0 is a maxterm. So, this corresponds to the maxterms M naught, capital M0, which is the same as w plus x plus y plus z. You can see that by doing a logical OR of this with this. So, w plus x or y plus z, that is the term 0000. This term corresponds to the row w plus x and the column y bar plus z. So, this is w plus x plus y bar plus z. This term corresponds to w plus x bar plus y plus z and so on. So, if I, if I look at this term, this is w bar plus x bar plus y plus z. So, this corresponds to 1100. So, you must see ((Refer Time: 16:35)). So, this is w bar plus x bar plus y z. So, these are all the different maxterms. So, 1100, oh sorry, I was looking at a 1, I should look at a 0. Let me pick a 0 here, this corresponds to the term 11and 01. So, this is, this is 0. So, 1101 should appear here, 1101 is appearing in the list.

So, now, let us see how to combine them. The same rule, as we did earlier, apply except that we are going to group 0s. We are still going to group in powers of 2. We can only group adjacent cells and for products of sums you can only group 0s. Let us see the various grouping that are there.

(Refer Slide Time: 17:19)

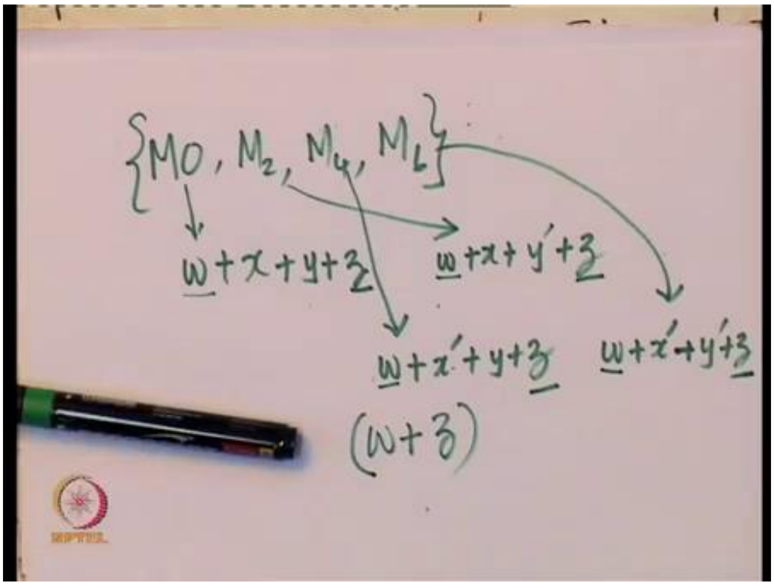


So, this is the grouping because these are four 1s that are adjacent to each other. So, it is, it is a square that you are looking at, right. There are several other groupings. For example, M4, M5, M6, M7 also forms a group. So, this is, so let us look at the terms here. M4, M5, M6 and M7, that is, this row here, right. Let us look at this one. M6, M7, M14, M15. So, this is M6, this is M, sorry, this is M6, this is M7, this is M14, this is M15. These four 0s are also adjacent to each other. So, that is one of the terms and so on. So, in this picture I am only showing one of the grouping, I am not showing all the groupings. So, this is to get all the, this is to get the possible groupings.

Now, how do we get a simplified expression? We have to cover all the 0s and we want to cover it as few time as possible. This is the thumb rule, we want to cover all the 0s, but we want to cover it with as few groupings as possible and as large grouping as possible at the same time, right. So, fewer, but large ones. So, let us look at the grouping here. If I started with this one, so I have, I have these four 0s taken care of. I still have these four 0s and these four 0s. So, now, let us say, if I group these four together, I will still have these two left out. So, I cannot expand these four into this column because I have a one here, otherwise if this was a 0 I could have taken all these two rows together, combined them into one term. However, I cannot do that now because there is one here, I can only group these four remember, I cannot group 6 of them together, that is not valid. I can only group in powers of 2. So, in this case I can group these four together and I will still be left out to these two 0s here, but they are in this column containing all the 0s. We can group all of them together also and we get various terms. So, this grouping, right, is M0, 2, 4, 6. So, this is 0, this is 2, this is 4, this is 6.

So, we can do this grouping is this sum term and this grouping here corresponds to this sum term and this grouping here corresponds to this sum term. If you notice, all the 0s are covered at least once and we have tried to cover it with as larger a group as possible, which is still a size of power of 2. So, these are the three different sum terms that we get, M naught, M2, M4, M6, M5, M7, M13, M14 and M2, M6, M10, M14, these are the sum terms that we get. Now, we can go and write the expressions for this directly by looking at M naught, M2, M4, M6.

(Refer Slide Time: 20:21)

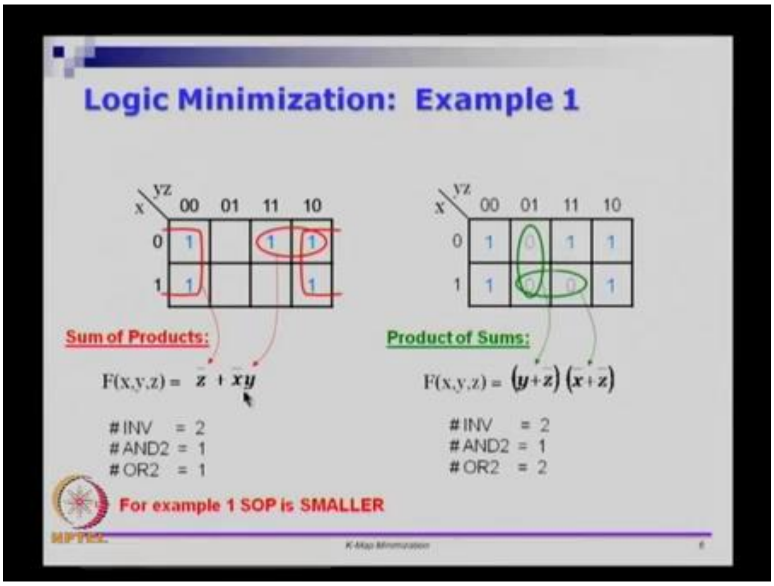


So, let me show you how to get the corresponding expression for M naught, M2, M4, M6; M naught, M2, M4, M6. So, the product term, that, so remember this is the set, it is not a product of these four. So, the sum term that corresponds to this should have this one, right. So, in some sense, you go and look at this one. So, this is w plus x plus y plus z, M2 is w plus x plus y bar plus z. M4 will correspond to w plus x bar plus y plus z and M6 will correspond to w plus x bar plus y bar plus z.

If you look at these four terms, right, you can notice, that what we are having is, this w is appearing in the same form in all the four. So, you can notice that here w is appearing in the same form everywhere and z is appearing in the same form everywhere. So, the combination of these four is actually just w plus z, right. You can infer that by coming back looking at the table here. So, these two rows have w in common. So, the sum term should have w. If you look at these two columns here, which this has z in common, this y plus z in y bar plus z. So, these four cells have w plus z and similarly, these four cells we have x bar in common and in these two columns we have z bar in common. So, these four cells correspond to x bar plus z bar.

Finally, the last column here, w and x are continuously changing, but this corresponds to the term y bar plus z. So, this one is w plus z this one is x bar plus z bar and this last column is y bar plus z. You write that as a simplified Boolean expression w plus z into x bar plus z bar into y bar plus z. In this example we have grouped the 0s together and we have got a product of sums.

(Refer Slide Time: 22:39)

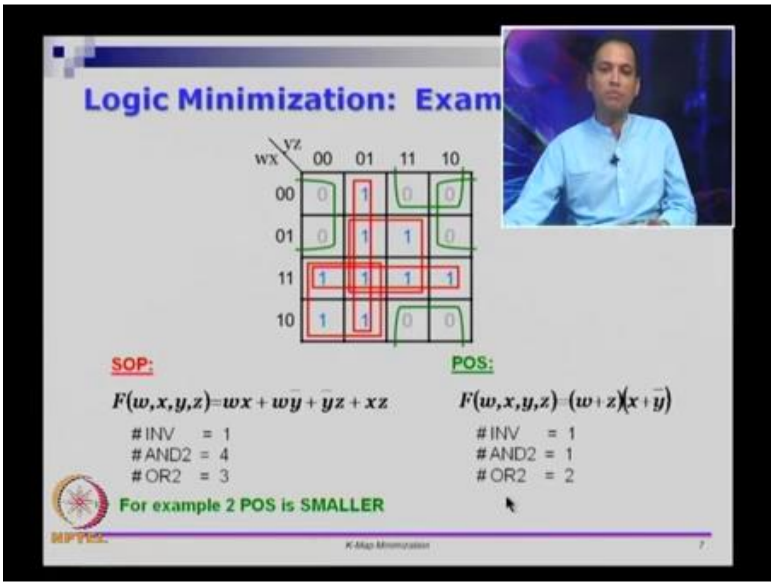


So, let us look at some, some more examples from here. So, if I, if I have a table like this, right, this is the expression where we have, it is a function of x, y and z. Let us assume, that x is the leftmost variable and y, z appear in the next two columns in the truth table. If I want to group them together, right, the sum of products will be grouping all the 1s together. So, we have a group of four and in this group of four it corresponds to z being 0 as a column. So, it is z bar and this 1 is still left out, I have to group that also when I do that I get the term x bar y because it is in the row x complement, but it is across to columns where z is flipping, but y is constant at y, right, this x bar y. So, z bar plus x bar y.

If we want to do same thing in product of sums, you group two 0 together. So, if group these two 0s together, that will corresponds to the term y plus z bar and if I combine these two together, it corresponds to the term x bar plus z, z bar, sorry, it is z bar because z is 1 here. So, this row corresponds to x bar, these two columns correspond to z bar. So, it is x bar plus z bar. You can actually go and take this expression and simplify it, you will end up with the expression on the left side. So, you have product of sums and sums of product, right.

In this example, if you want to implement this using gates, this z bar plus x bar y bar x bar y will require 2 inverters because for z and x you need inverters, there is 1 AND gate here and 1 OR gate here. So, we have 2 inverters, 1 AND, 1 OR gate. So, this 2 here implies, it is a two input gate. If you do the product of sums, we need 2 inverters, 1 for z bar and 1 for x bar. We need 1 AND gate to do this product, but we need two or two input OR gates, one for this term and one for this term. So, in this example, right, it looks like this one needs 4 gates and this one need 5 gates. So, for example, one the sum of product expression, this way of simplifying is better rather than this way of simplifying it.

(Refer Slide Time: 25:04)

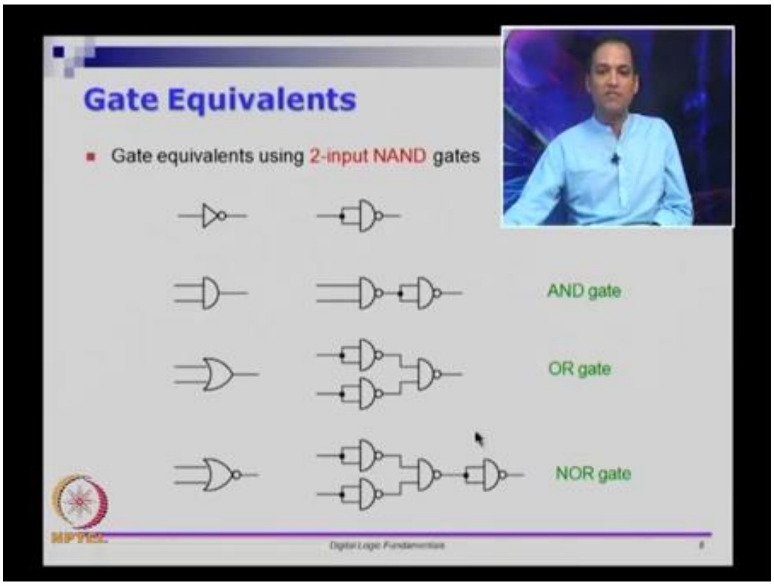


Let us look at one more example. So, this is an example in which we have four variables, w, x, y and z. Let us group the 1s together and get sum of products first. So, so let me show you the one which is not grouped. We can see, that there is group of four 1s here, there is the group of four 1s here, there is the group of four 1s here, there is the group of four 1s here and so on. There are several groups of four here. So, let us pick all the things that we need. So, there are four groups of four here. So, this term corresponds to w being 1 and y being complemented, so you have w y bar. This column here corresponds to y being 0 and z being 1. So, you have y bar z. This row here corresponds to w being 1 and x being 1. So, you have w x and finally, this group of four corresponds to x being 1 and z being 1. So, you have x z. So, f of w, x, y, z is w x plus w y bar plus y bar z plus x is z. So, this is the simplified expression for this truth table that is given here or the K-map that is given here.

So, if you do the product of sums, the product of sums will require you to group the 0s. So, there is a group of 0 here and group of here and that is two groups of size two. So, it makes the groups of four and there are two groups of size two here, which can be combined to make a size of four. So, you look at this term here and this term here, right that corresponds to the fact, that w is 1 and these two columns correspond to the fact that z is 1. So, the intersection of that gives you w plus z. And if you look at these, this grouping here, this corresponds to the fact, that x is 1 and these two columns. So, these two rows correspond to the, so the first row and the last row correspond to the fact that exist one and the last two columns correspond to the fact that y is 0, the intersection of that gives me x plus y bar. So, this is the sum of product and this is the product of sum.

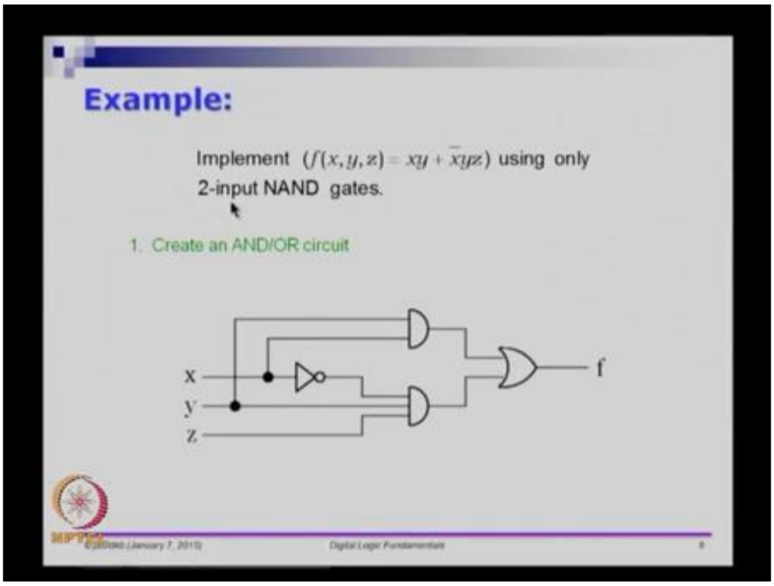
In this case, you will need one, two, three, four AND gates, one, two, three OR gates and one inverter, right; only for y you need an inverter. So, 1 inverter, 4 AND gates and 3 OR gates, whereas in this one you need 1 OR gate here and 1 OR gate here, 1 AND gate here and 1 inverter there. So, 1 inverter for y bar, 1 AND gate for the product and 2 OR, 1 AND gate for the product and 2 OR gates for the individual terms. So, in this example, it so turns out, that the pos is smaller, upfront you may not be able to say whether sop is better or POS term. In this example POS is better, in the previous example SOP form was better.

(Refer Slide Time: 28:11)



So, let us wrap this module with sum equivalences, we already did this. So, inverter is equivalent to handing the inputs of the, so connecting the same input to both the inputs of a NAND gate and AND followed by an inverter. So, NAND followed by an inverter is AND. If we invert the inputs and give it to a NAND gate that is equivalent to an OR gate and if you attach an inverter to the output of this circuit here, that gives me NOR gate. So, this is just for you to recollect what we did so far.

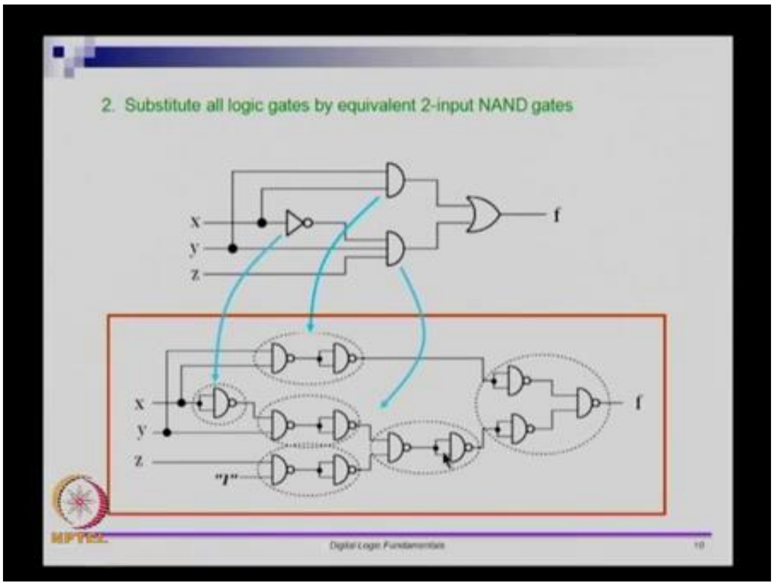
(Refer Slide Time: 28:47)



So, let us see we want. If we want to implement a function like this using only two input NAND gates, right. So, x y plus x bar y z, you want to implement this using only two input NAND gates, then we can do something like this. So, this is x y, we AND it together, you have it here, then this x bar y is z AND it together and you get, you logically OR it. So, this AND/OR circuit, right. So, you have the inversion for the literals and then you have a set of AND gates, which give you the product terms and then you put in OR gate, that gives you the sum that gives you the sums.

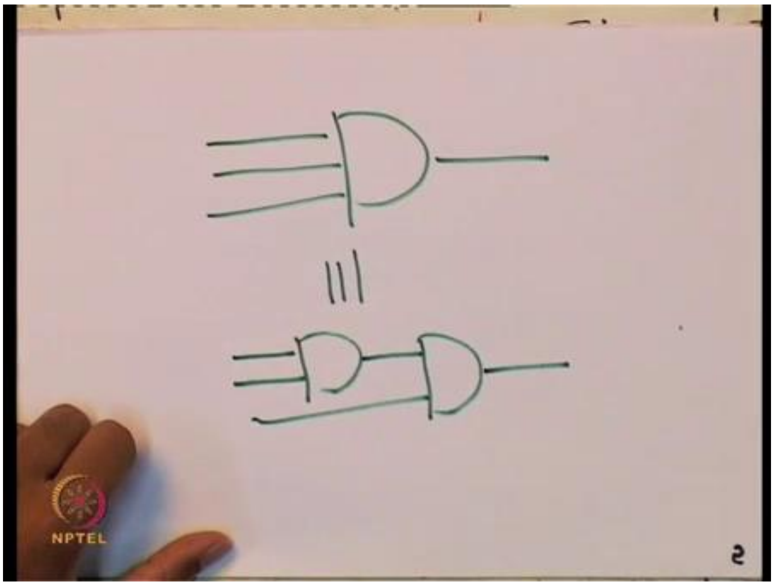
So, you are given a sum of product and you have the circuit implemented using AND, OR, NOR, but the question is asking you to implement this using NAND only and not just any NAND, it is asking you to implement using only two input NAND gates. Let us see how to do something like this. So, the first thing we are going to do is substitute all the logic gates by equivalent two input NAND gates.

(Refer Slide Time: 29:44)



So, we will see. So, if I take this circuit, I substitute that with two input NAND gates, that is what the circuit that you have in the bottom is, what we will get, let us see how. So, this inverter is here, this AND gate is NAND followed by an inverter. So, that is there. And this three input AND gate is actually written as a combination of two, two input AND gates. So, this circuit is actually written up as the combination of two to input AND gates. So, you can see, that there is NAND followed by an inverter and there is another NAND followed by another inverter. So, it is essentially a cascade of two AND gates. So, if we want a three input AND gate, you can do something like this.

(Refer Slide Time: 30:47)

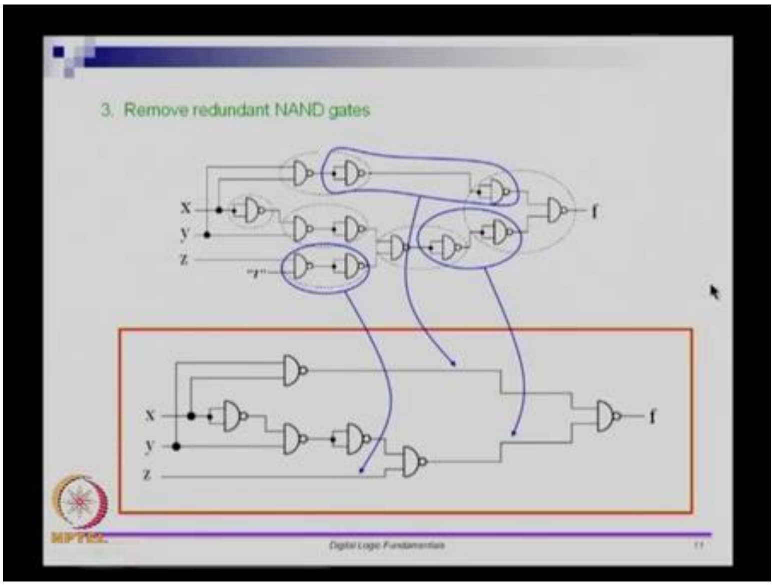


So, let see how to get a three input and gates. So, if I want a three input AND gate, this is equivalent to writing it as one 2 input AND gate followed by another two input AND gate, right. And to get this AND gate we have a NAND followed by an inverter to get this AND gate. We have a NAND followed by an inverter that is what you see in the picture here, right.

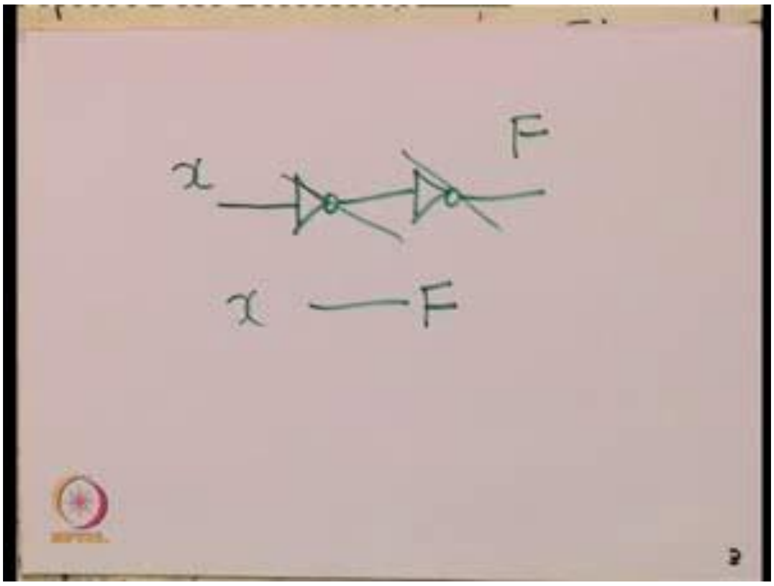
So, coming back to the slides. So, you can see, that this NAND followed by the inverter is one of the two input AND gate and this NAND followed by this inverter is taking care of the other inputs. So, it is a cascade of two, two input AND gates. And finally, this is how, this is an OR gate. You can see the structure here. There are two inputs, which are complemented and is going to a NAND gate. So, that means, it is an OR gate. So, what we have now is, we have taken a AND/OR circuit and drawn it using only NAND gate. So, in this picture you can notice, that there are NAND gates and nothing else.

So, the only extra things seems to be this for z instead of taking z and putting directly here, we have it as z and one complement, which is z bar and complement of that, which is z. So, we have z, instead of giving a line directly to this NAND gate here, we have it as a NAND with one followed by an inverter that is same as z itself. So, this has only two inputs NAND gates and nothing else.

(Refer Slide Time: 32:23)

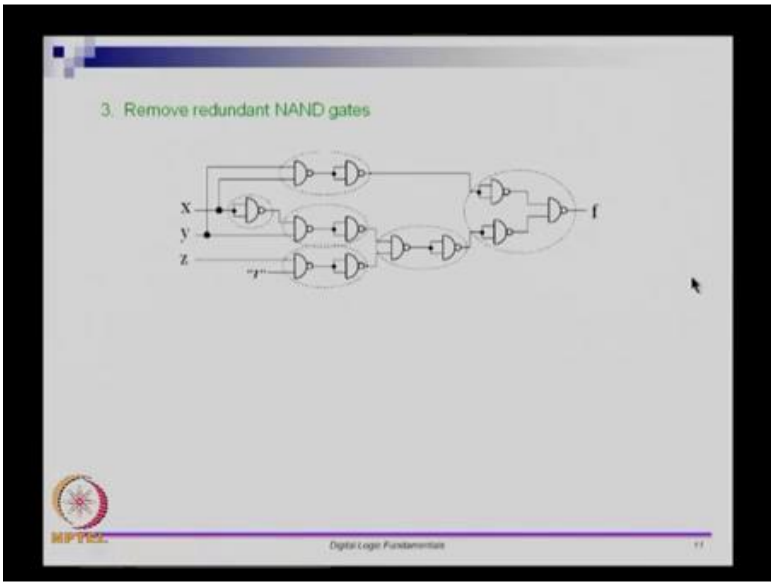


(Refer Slide Time: 32:34)



Now, this is not most simplified form. One thing we can do is, remove redundant NAND gates. So, when is something redundant? So, let us see when something might be redundant? If I have a circuit in which I have an inverter followed by another inverter. Let say I give you this as x and this as F, an inverter followed by an inverter is actually redundant. Two inverters actually cancel their actions. So, this is equivalent to saying x is F itself. Is it not?

(Refer Slide Time: 32:55)



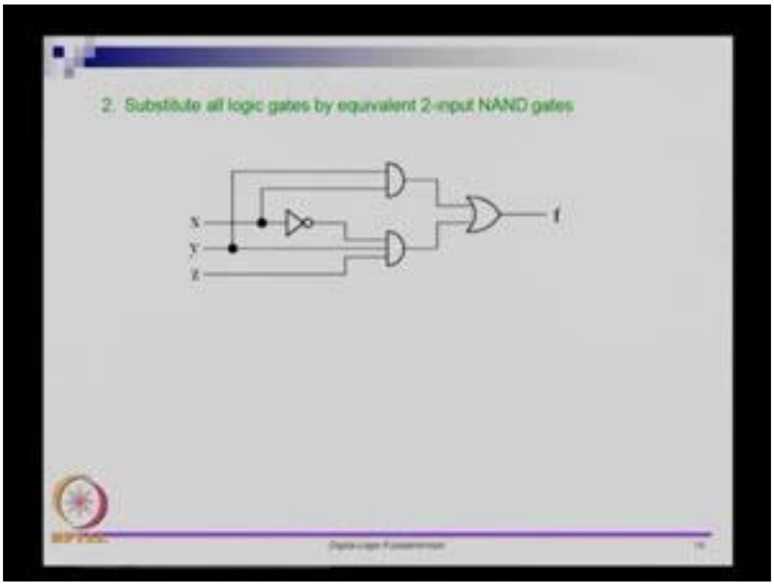
So, the same thing that we can do in the NAND gate also. Whenever we see an inverter followed by another inverter, we can remove those two inverters and connect them with the wire. So, in this picture if you notice this inverter, right, this is an inverter and this is an inverter, but implemented using a NAND gate. So, this is the function, which is doing inversion. This is also another gate, which is doing inversion. So, I take inversion and inversion here, inversion of inversion is just the wire itself, I can remove that wire and there are other places where you can do this same thing. So, this is an inversion, this is an inversion I can remove the two inversions together and get rid of it.

Similarly, there is an inversion and this is an inversion, I can get rid of that also. So, this wire corresponds to the these two inversions being get, being redundant, being get ridden off, then this wire is the same as these two inverters and getting rid of one. And finally, these two inverters when you get rid of the wire. So, in this circuit we have a 1 plus 6, 7 gates, plus 2, 9, plus 3, 12. So, this direct implementation has 12 two input NAND gates, whereas once it get rid of the inverter is, we get rid of three inverters. So, this gives us a circuit with 9 two input NAND gates. So, this is better than the circuit that you have here.

So, if you notice this circuit this circuit is actually doing the same thing as the AND/OR circuit that we did before. It is actually doing the same thing as this circuit here with the redundant gates, only that it is simplified. It has much fewer gates than this one here and it satisfies the problem statement that we started with, implement using only two input NAND gates, right.

This brings me to module 9 of this week and what we saw in this module is, we saw two parts. One part in which we did in some minimizations using the truth tables, and the second part of the module in which we saw the writing circuits in terms of a specified structure of the gate. So, specifically we looked at using only NAND gates. I, I will give this to you as an exercise.

(Refer Slide Time: 35:13)



Take this circuit that is given here, the circuit and implement that using only two input NOR gates and figure out the number of gates that you need for it. What is the smallest number of gates that you need for implementing this circuit using only two input NOR. Go and think about that. So, this brings to me the end of this module. So, we have a few more modules to look at for this week. So, I will see you in a little while.

Thank you.