Digital Electronic Circuits
Prof. Goutam Saha
Department of E & EC Engineering
Indian Institute of Technology, Kharagpur

Lecture – 10
Karnugh Map and Digital Circuit Realization

Hello, everybody. In the last class we had seen how to convert a truth table to a Boolean expression; for which we had seen minterm based SOP, sum of product presentation representation and maxterm based product of sum representation. And, we are also seen how to realize it using AND bank and a finally, OR gate for SOP realization and OR bank and a final stage second level OR gate for POS representation. And, also we had seen NAND-NAND and NOR-NOR equivalent circuit for them.

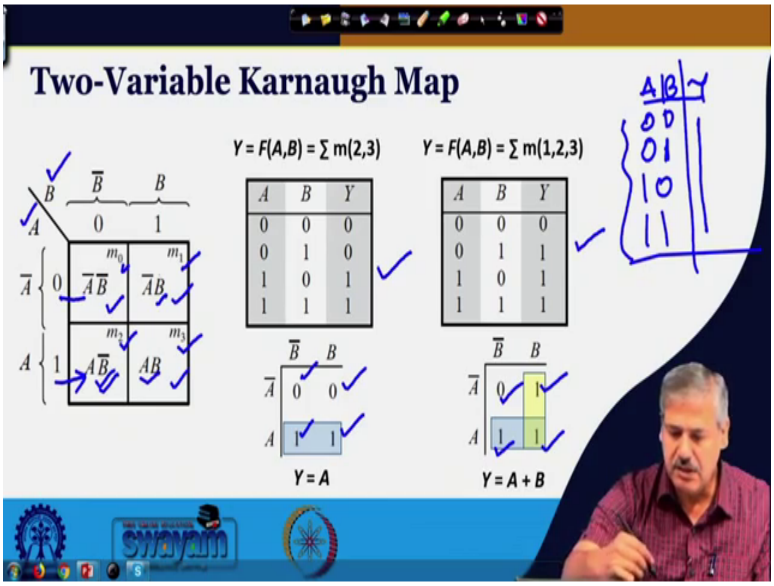
Now, if you want to have a more simplified or minimized realization of a truth table do you need to go for algebraic simplification of the minterm based or maxterm based expression that you have got? Or we can we have a better method.

(Refer Slide Time: 01:13)

So, Karnugh map gives us a very convenient method to simplify a SOP or a POS based true to get a SOP or PO based POS based expression. So, we shall discuss this thing in this particular lecture how to simplify. First we shall look at the how to represent a truth table in the form of a Karnaugh map and then we shall also look at use of don't care in

Karnaugh map simplification taking upon example and then we shall look at dual circuit and self-dual function.

(Refer Slide Time: 01:47)



So, in the Karnaugh map based representation what we do actually; the truth table we present in a different manner. The truth table we had earlier what you have seen that there are A if it is a two-variable truth table. This is Y. So, 0 0 0 1 1 0 1 1 4 possibilities are there and corresponding function output that you see over here, ok.

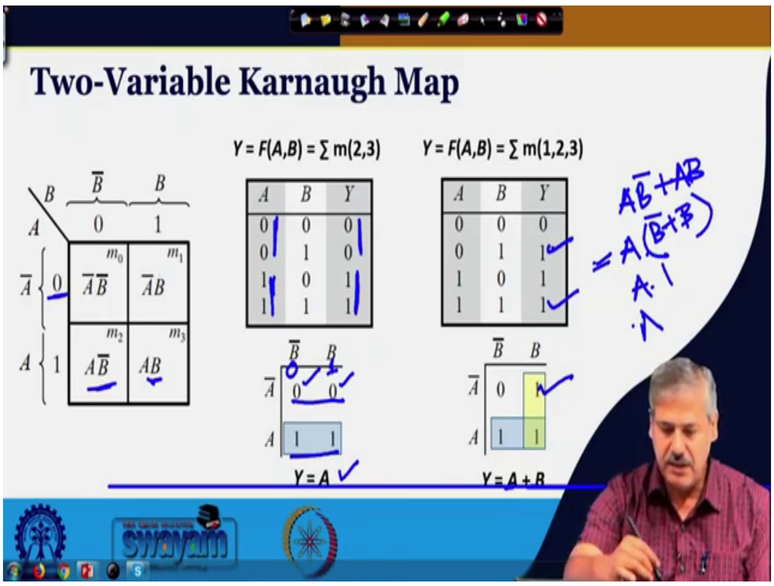
So, in the Karnaugh map presentation for a two-variable case we are having a rectangular space like this, where this side is variable A and this side is variable B. The values represent 0 and 1 for B this column represents always B remaining 0 and this column represents B always remaining 1. And this row represents A remaining 1 this row represents A remaining 0. So, at the cross point you have got a combination or specific combination of A and B. So, this particular place is a combination of 0 0, this is 0 1, this is 1 0 and this is 1 1.

So, all the four thing that you see in the truth table are represented in the Karnaugh map in their respective location, ok. And, in terms of min-term, if you look at it in terms if minterm then each of this spaces if there is a presence of 1, so, A bar B bar will be the corresponding minterm, A bar B will be the minterm over here, AB bar will be the minterm over here and AB will be the corresponding minterm. And, also we also already know how to represent a minterm in terms of it is decimal suffix. So, this is your m 0, this is your m 1, this your m 2 and this is your m 3, clear?

So, this is how we look at it two-variable representation of a Karnaugh map and this will be extended for three and four variable multi multivariable cases. And, if we a look at the truth table that we had seen before that is over here before that the another truth table which is the simpler only two 1's are there. So, the corresponding output you can see that the function output 0 0, this is 0, 0 1, this is 0 output is 0 1 0, this 1 and 1 1, this is 1, right. So, this is what we can see as going as a mapping and the one that we had seen before 0 0 it is 0 and rest of the three 0 1 it is 1 1 0 this is 1 and 1 1 this is 1, this we have already seen. So, this truth table when gets map to Karnaugh map it would looks something like this, ok.

Now, if you look at this representation and the kind of logical connection between input and output what we can see that in this truth table it is evident that the output is 1, output is 1 when A is 1 and output is 0 when A is 0 irrespective of value of B, isn't it?

(Refer Slide Time: 05:09)



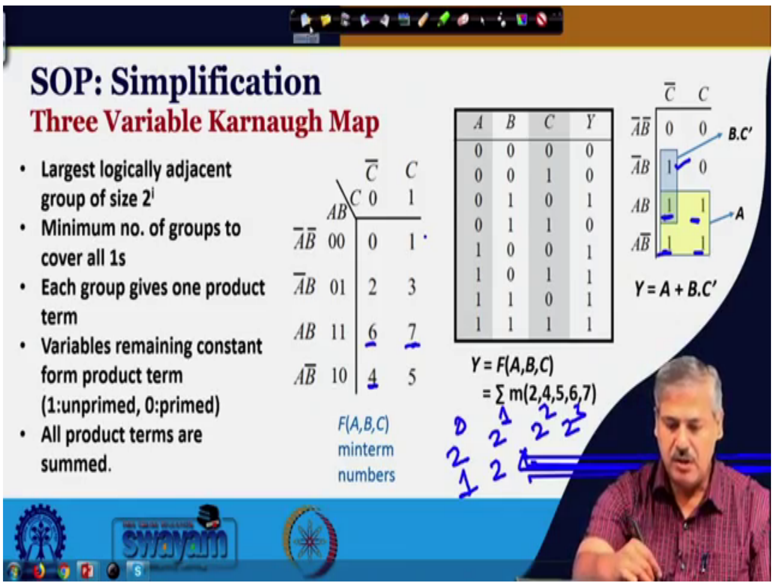
You can see that A is 0, output is 0. These two cases and A is 1, output is 1. So, you can see the corresponding when A is 0 this was the 0, A is equal to 0 case, right. So, both are 0 and A is equal to 1 both are 1, ok. And so, from the truth table we can see that the Y follows A. So, we can write Y is equal to A. So, which is nothing, but where this row A remains at value 1, irrespectively of the B changing from 0 to 1. So, B bar means here it is 0, B means it is 1 respective of the B changing for value A remaining on output is always Y at 1.

So, we can see and output is 0 for a remaining 0. So, we can write Y is equal to A by looking at these two value which can be put together grouped together. So, you look at this any presentation. So, again this 1 1 is present, right. So, this particular group can give us whatever if this 1 was not present right A minimized version of A which had been the case otherwise as AB bar plus AB this two minterms AB bar and AB and we can you know it is well understood that from complementarity we can get A some problem, ok. So, this is A, right.

So, this A comes over here, right and if your this was not there only B this particular these two 1's were present; that means, this one and this one right ok. So, this is the case when B is these two cases are present. So, you can see that when B is 1, the output is 1, right and these two can be grouped together and by which if in absence of this we would have got a B, Y is equal to B. So, these two can be summed together and we can see that there is a I mean these two can be put together and we get AY plus BY or B representation.

So, this is something which we can compare with what we have done in the past and the basic relationship and the truth table. But, can you have a generalized rule or method by which we can group these things? We do have such method, ok.

(Refer Slide Time: 08:11)



So, for which we look at SOP simplification rule which is taken taking forward this discussion. So, for that we look at first a three variable representation of the Karnaugh map. So, for the three variable representation of the Karnaugh map we have got three variables A, B, C.

So, on one side you have got C at 0 and 1 which is similar to what you had seen before and this side we have got AB. So, four possibilities are of AB 0 0, 0 1, 1 1 and 1 0. So, now, you remember that after 0 1 you are not going to 1 0 we are going to 1 1 because we are looking for a logical adjacency between two neighborhood positions, two neighboring position, two adjacent positions, which will help us in forming groups, right.

So, by logical adjacency we what we mean is that between two positions only one value will be changing one of the A, B, C value will be changing. So, this particular position you see 0 0 1, this particular position is 0 0 1, A bar B C bar and here next one is sorry this particular position is 0 1 0, next one is 1 1 0. So, only A is changing from A bar to A. So, if it was 1 0 then two values would have changed A would have changed as well as B would have changed which we are not allowing here to have a better opportunity or better way of convenient way of forming the formation of the groups. This we need to take note of when we form the Karnaugh map for three variable, the other side is the same.

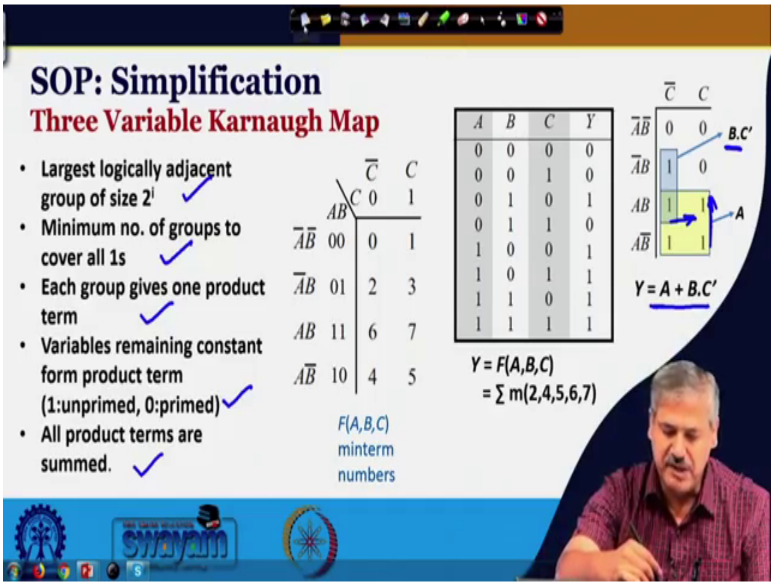
Now, if you look at one example. So, this is the example of a function where the minterms are 2, 4, 5, 6, 7 and we can here again the corresponding minterms the m 0, m 1 we are not writing in this manner we are just writing the decimal equivalent of 8. So, 0 0 0 this is 0, this is 0 0 1 is 1, then 2, 3 because AB bar 1 0 is coming over here so, 4, 5 is here, right. And, then 1 1 0 is 6 and 1 1 1 is 7. So, the minterms are like this. This was the minterm suffix mention of those thing. So, here 2, 4, 5, 6, 7 so, 0 1 2 2, there is a 1 present over here, right and 4 5 6 7, 4 is here, 5 is here, 6 is here and 7 is here, ok.

So, that way the truth table is converted to three variable Karnaugh map, is it clear? Right. Then how we found the group? We will look at largest logically adjacent group of size which is in integer power of 2. So, integer power of 2 is what? 2 to the power 0, 2 to the power 1, 2 to the power 2, 2 to the power 3 like this, ok. So, that means, size of 2 1 2 4 8. So, these are the different things that you are looking for in while forming the largest group, ok. So, and they should be logically adjacent, right and it in the form of a rectangle like this which is very clear here and while talking about logically adjacent more we shall discuss little later.

So, the groups that we can form here you can see this is the group of 4, largest group of 4 that we can form and this is another group that we can form over here which covers all the 1's, right. So, in the SOP realization all though 1's need to be covered, and each group will give us one product term the way you had seen in the previous example. So, one was giving A and other was giving B and we had a summation of A and B to get Y is equal to A plus B for the second example for the two variables we have seen before.

So, here this particular group when you are looking at so, the product term that will come out of it we shall see which variable is remaining constant within the group; that means, which is not changing, right.

(Refer Slide Time: 12:57)

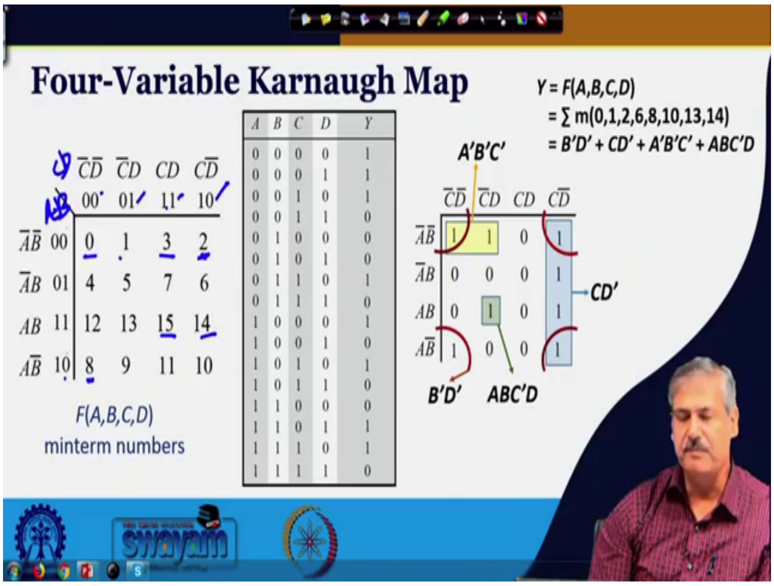


So, if you see within this group C is changing from C 1 to C along this direction and along this direction B is changing from B to B bar B bar to B right. So, only A is not changing. So, this product term will have only one variable which is A and other variable since it is changing it is not to be there. So, this will be form one product of and this is another group. In this group again we shall see which is not changing. So, we see that B is not changing right and C is not changing and remaining with the value C bar. So, this will generate a product of BC bar, right. And, we know that for this particular group A is changing from A to A bar.

So, together when you sum it up we get A plus B C prime that is the SOP realization of this particular truth table ok. So, whenever so, if you put it in the form of a rule; so, largest logically adjacent group of size 2 to the power i that is integer power of 2 we shall see, minimum number of all groups to cover all ones, ok. Each group will give one product term. Variables remaining constant form the product term that we have said and if it is variable remaining constant with a value 1, ok, it will be unprimed and if it is 0 it will be primed, ok. So, very well remaining constant here with say 0 C prime. So, we will go there and B B over here remaining with 1, so, only B is going as unprimed or un complemented and all product terms are finally, summed clear.

So, this is the way we shall form the SOP with the Karnaugh map based simplification.

(Refer Slide Time: 14:41)

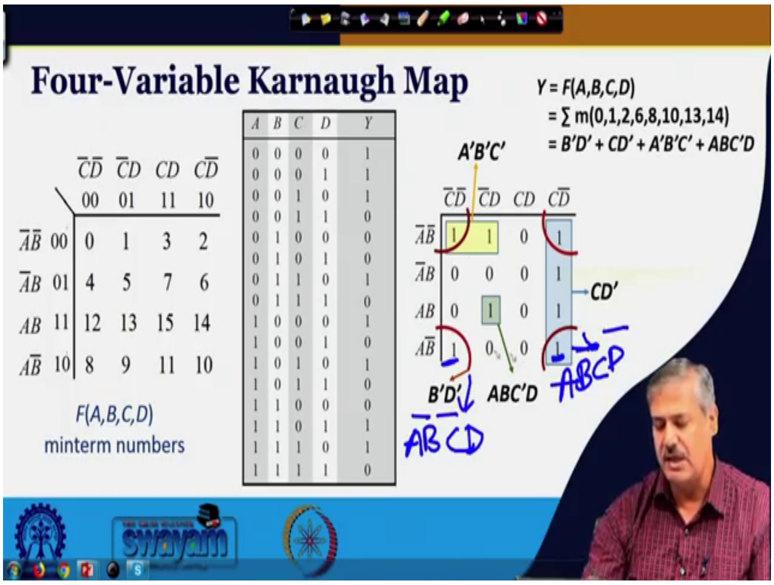


Now we can extended to four-variable Karnaugh map, ok. So, for four-variable Karnaugh map we shall be having a rectangular space where four 16 different possibility, 16 different locations will be there, ok.

So, you can see one side is AB, another side CD. So, this side is AB and this side CD. So, AB is having value 0 0 0 1 1 1 as I said because you want logical adjacency after that 1 0, right. Similarly, for CD 0 0 0 1 up to the 2 1 1 because of the need of the logical adjacency and then there is 1 0, ok. So, the corresponding minterms the suffix the decimal equivalent will be 0 1, since it is 1 1 here so, 1 0 over here. So, 2 comes after 0 1 that is a is 2 over here and then 3, 4, 5, 6, 7 here because of 1 1 presents present here 8, 9, 10, 11 and 12, 13, 14 and here there is a 15 ok. So, that is the four-variable Karnaugh map the minterms their where their take position.

So, if you take this example so, in these example we have got minterms 0, 1, 2, 6, 8, 10, 13, 14 just one example.

(Refer Slide Time: 16:17)



So, 0 over here, 1 this is 0, 1, 2 right then 6, 6 this is 3, 4, 5 this is 6, ok. Then this is 7. So, this is 8, 9 this is 10, ok, then this is 11, this is 12, right and this is your 13 and this is your 14, ok. So, all the minterms are represented in the truth table in the Karnaugh map. So, you can see these their presence now you have to form the largest group of size 1, 2, 4, 8 I mean depending on how you can associate logically these groups.

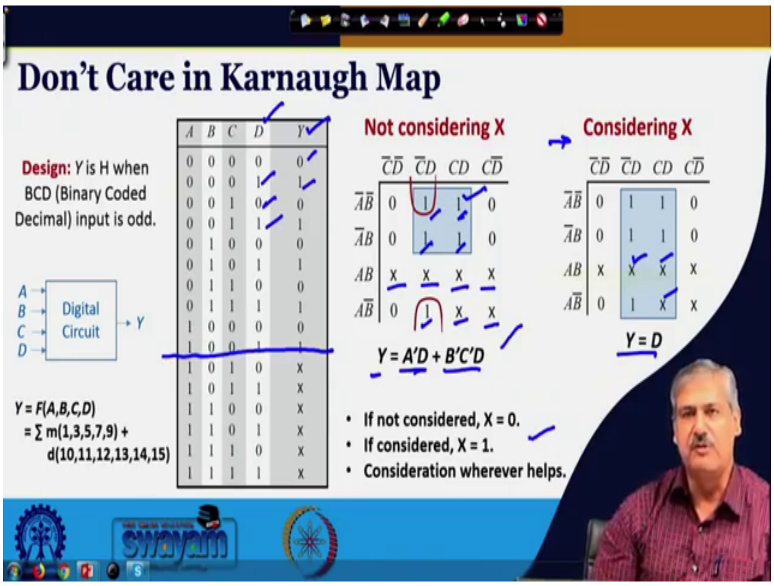
So, what we can see these four ones can be group together, right and how what product it will generate? The variables that are remaining constant; so, C and D are remaining constant C with the value 0 and C with the value 1 and D with the value 0. So, CD bar will be the corresponding product term, right. And, A are B A and B are changing along this group, ok. Then the other group that is possible we can see over here this is the other group this yellow you know background. So, for that A bar, B bar and C bar; so, this is what is remaining constant, and D is changing from D to D bar over this particular group. For this is one it cannot be grouped with any other. So, this is all the variables come into picture ABC prime D, AB C prime D. So, AB from this side and C by C prime D from the column side, ok.

Now comes interesting part which I was saying that while these are evident we need to take note that the one that are in the edges. So, in the edges say this particular one, this is this location is AB bar, C bar, D bar and you look at the other one this is this is A B bar C D bar and this one is A B bar C bar D bar. So, this is this location, this is this location, ok. Now, between these two locations what do you see that A B bar C bar. So, this is A B bar C bar and only the D is changing from D to D bar, ok. So, only one variable is changing. So, though they are physically not adjacent, but they are logically adjacent. So, if nothing else is there only this one and this one is present; that means, this one and this one is present we can form a group of these two.

And, what will be remaining constant what term it will be there A B prime C prime because D is changing. So, that will be the corresponding term. So, that way all these four corner terms B one that is shown with these you know brown this thing they can be grouped together and the term that is remaining constant for them you can see is B prime; B prime is constant for all these corners right and also D prime is constant. So, B prime and D prime is other term. So, this is valid for three variable and you know problem as well if we look at those Karnaugh values we can see a logical adjacency.

So, ultimately the four terms that we product terms we sum it up and we get the corresponding Karnaugh map representation.

(Refer Slide Time: 19:53)



So, now we look at the use of don’t care in the Karnaugh map, ok. So, in the what do you mean by do not care? So, some of the input combinations which is of no interest to us; for example if you have a design problem where the input to a circuit is binary coded decimal; that means, only decimal number of coded with binary. So, decimal numbers input is 0 to 9. So, the binary coding is 0 0 0 0 to 1 0 0 1 these are the inputs. So, at for that whenever the input is odd, ok; that means, 1, 3, 5, 7, 9 the output will be high otherwise the output will be remaining low.

Now, for this code variable inputs when you from the you know Karnaugh map and all. So, there are 16 possibilities. So, beyond 0 to 9; that means, 10 to 15 the decimal equivalent you do not have you know you do not need to care because those inputs will never come. Input is mentioned as a BCD input binary coded decimal input. So, we write in SOP this in the minterm with representation. So, m minterms 1, 3, 5, 7, 9 and these are the don’t care 10 to 15 are the do not care. And in the truth table though they go as a X; X means it could be you know either way, 0 or 1 you don’t care actually ok.

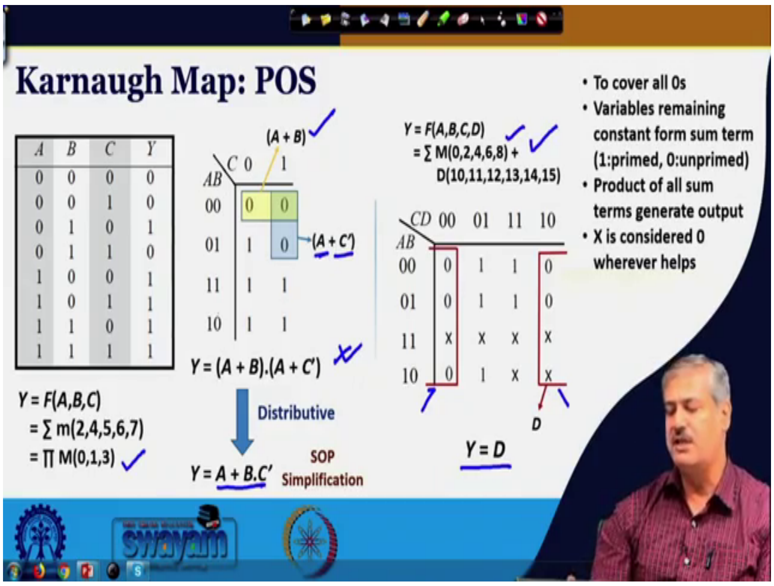
So, in the Karnaugh map mapping how it till reflect? So, in the Karnaugh mapping so, wherever your this minterms 0 1 1 2, this is 3, this is your 5, this is your 7 and this is your 9, those 1’s will be present rest are 0 up to 0 to 9 and 10 this is your 8, 9 this is 10, 11, 12, 13, 14, 15. So, these are all do not cares, ok. So, now how you form the largest size group; so, to start with if you do not know how to handle the don’t care and all you are a bit you know circumspect.

So, you may ignore them and you just try to form largest group possible with the help of with the two covered all the ones. So, in that case the largest be possible over here is this is one and this is another one that I said to the edges they are logically adjacent and you get for from this one A prime D and the other one you can B prime C prime D. So, this is the expression that we will get. But, when you do that effectively what you are doing you are considering this don’t care as 0 is it not? Since you have not covered it if you just put the value say 1 0 1 0 that is your decimal 10 the output here generated will be if you just substitute those values Y will become 0, right. So, effectively you are treating them as 0, while we have the option of treating it as 0 or 1 because you do not care, right.

So, if you consider the X as don’t care that you can include consider it as one if it helps you in forming the larger largest larger group size of integer part two, then you can go ahead with that when you doing that you can see this three don’t care condition can be considered as 1 and a larger group size can be formed. And, one particular group can cover all the ones, because there is the necessary that all the 1’s need to covered and in this particular group you can see that only D remains constant and rest of the things AB and C changes. So, Y becomes D, ok. So, whenever it is convenient we shall use don’t care as 1; that means, if it helps in making a larger group size. And it is also evident if you look at this truth table this is your D and this your Y, right.

So, you look up to 1 0 0 1 this particular case right and we see whenever D is 0 output is 0, D is 1 output is 1, D is 0 output is 0, D is 1 output is 1,. So, this is also we done from the way if you known the truth table is formed, ok.

(Refer Slide Time: 24:19)



Now, this can be used for POS; that means, product of sum representation also. So, for that we look at some example. So, the example that you have seen for the three-variable case 2, 4, 5, 6, 7, now for from maxterm point of view where 0s are to be covered, ok. So, the max term are 0, 1, 3. So, these are the maxterms that are need to be covered. So, these maxterms when you cover it you will find this is 0 0 that is remaining constant for this particular group. So, 0 for maxterm representation we know whenever 0 it will be go as uncomplemented and whenever it is one it will go as complemented.

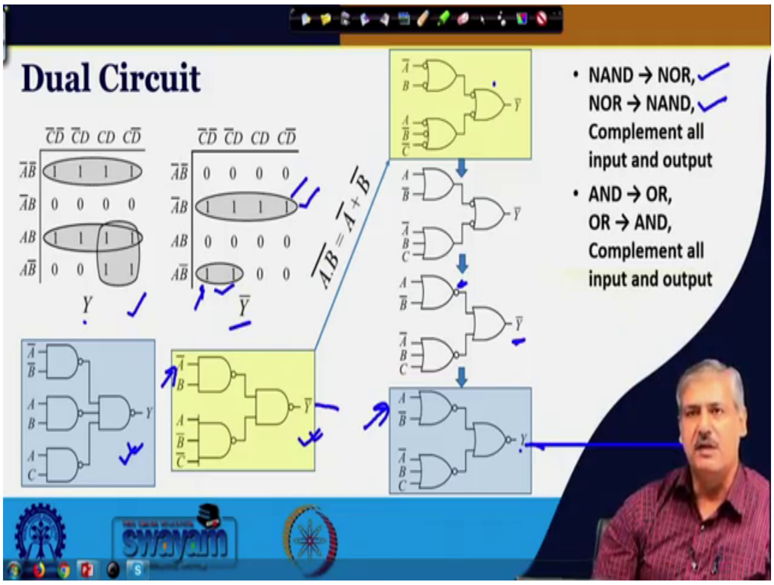
So, this is sum term that is from here is A plus B and in this case A is remaining constant with 0. So, A will become A go uncomplemented and B is remaining constant with 1. So, sorry C is remaining constant with 1 so, C will go complimented C bar ok. So, these are the two sum term which will if you just end it you will get the minimized POS representation. And, if you look at from distributive law if you apply you will get A plus B plus C prime which we got for the SOP based minimization where we considered all the ones,.

So, what we saw for a don't care condition so, similar thing can appear for this don't care condition in the POS representation also. So, the same example that we had seen before; so, the 0s that represents the maxterms so, 0, 2, 4, 6, 8 even positions they are zeroes. So, that is the those are the maxterms and 10 to 15 are the don't care values. So, this is what you see over here this 10 to 15 remains the don't care value.

Now, you have to form the largest group of zeros, ok. So, here again at the edges we have logical adjacency. So, the largest group that can be formed is by consider by considering this and this and over here what remains constant? You see that only D remains constant with a value 0 and for SOP POS representation the remaining constant with the value 0 means it is uncomplemented version will come in to the sum term that is formed and only one sum term is formed I mean there is no other term and other variables. So, X is equal to D, ok.

So, the rule says to cover all the all the 0s variables remaining constant from the sum term 1 is prime and 0 is unprimed, product of all sums terms are generated considered and next is considered 0 wherever helps; helps in forming larger sized groups ok.

(Refer Slide Time: 26:57)



So, a dual circuit is something which is useful that we would like to discuss in this context. So, the kind of circuit that we had seen before that was two-level in presentation consulting all the maxterms. Now, in the in that particular consideration we had seen for any SOP we can convert to NAND-NAND in any POS can convert to NOR-NOR irrespective of whether it is a minterm based representation or a maxterm representation by using De Morgan's theorem of first kind and first theorem and second theorem for two respective cases. This we discussed in the last class.

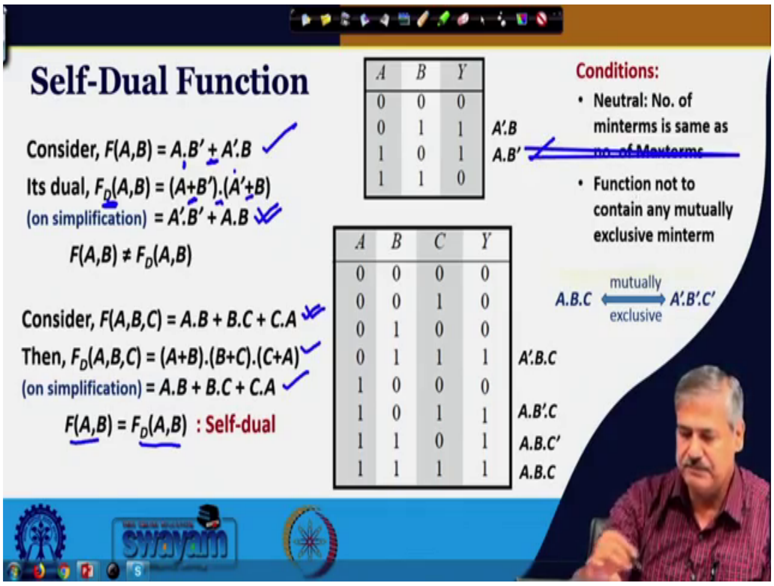
So, here also if we take up you look at this particular example. So, this is one particular truth table, ok. So, this is the corresponding NAND-NAND representation, that is fine. So, if you want to get this particular truth table the representation using NOR-NOR, ok. So, basically you are talking about POS representation, right. So, POS representation we have to consider all the 0s and we get the directly the NOR-NOR circuit from this by the discussion that we had before. Another way of looking at it is first you consider this was Y you considered Y prime, ok. So, Y prime is what its inversions. So, wherever there was 0, it will be then be 1. So, these are the cases.

So, the corresponding circuit that you get over here by considering these two groups A prime B and A prime B C prime D, is it not? This is A prime B C prime D. So, this is what you get and this is your A prime B, ok. So, from that you get the corresponding NAND-NAND circuit instead of AND, OR circuit. Now, if you look at this particular circuit NAND can be converted to from this your De Morgan's theorem a power A bar plus B bar, so that means, bubbled dot. So, this is what you are doing this NAND gate becomes a bubbled OR gate ok. So, this is NAND gate is also becomes a bubbled OR gate; that means, A bar these are in inverter that is put in the before the or gates. So, these inverter can be brought over here so, A becomes A prime becomes A and B becomes B prime, ok.

So, now, these bubbles over here can be brought to this place, ok. So, that this becomes an OR gate and finally, this output Y it was Y bar. So, if you take another bubble inverter of it becomes Y, ok. So, these becomes your POS representation by simply converting this particular circuit with the other circuit where NAND gates are converted to NOR, NOR will be converted to NAND and complement all inputs and outputs. So, these over the inputs, you see, ok. These inputs are complemented and complement all the output. So, this is also output that is getting complemented.

So, this is the way you can get a dual circuit. So, from NAND you can get NOR and NOR you can get NAND; that means, NAND is replaced by NOR, NOR is replaced by NAND and complete all outputs and inputs are complemented.

(Refer Slide Time: 30:07)



So, the last slide of this is a particular discussion. So, there is something called self-dual function. So, what is a dual function? Dual function is something where if you change the AND with OR and OR with AND what you get is the dual of the given Boolean function.

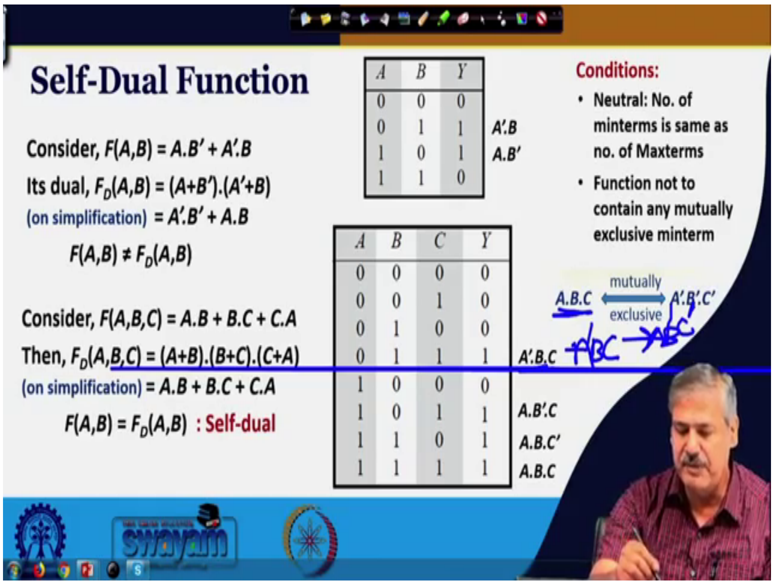
So, if this is the would function, then a given function it is dual represented in this manner is this place of AND you are putting a OR and in place of OR you are putting a AND, and this is AND. So, this is OR, ok. So, if you simplify then you get this one. So, obviously, you can say that these two are different. So, these are not same dual this is dual and the function is different which is possible ok, but if you look at another function say AB BC and CA, AB plus BC plus CA, ok.

This is the one and again you take dual of it; that means, change AND to OR and OR to AND. You get A plus B ANDed with B plus C ANDed with C plus A and if you simplify it you get back after the simplification stays steps the same function. So, this F D and this F AB they are self dual, ok. So, if you change the AND and OR there is no difference for such functions. So, when we can when a particular function becomes a self dual? So, is there is any rule? Yes, there is. So, for that let us look at their truth table.

So, if you look at the truth table of the one where it was not dual self-dual. So, this is what you get this is the one ok, sorry. There is some issue today anyway we can find out that the ratio these are the two minterms that are there and if you expand the this particular case the corresponding minterms are A prime B C AB prime C ABC prime and ABC, and this case a A prime B and AB prime right.

Now, one thing that is required for a function to be self dual is that it needs to be neutral; that means, the number of minterms will be same as number of max terms. What does it mean, that number of 0s in the function output will be same as number of 1’s in the function output. So, if it is a two variable case two 0s and two 1’s; three variable case four 0s and four 1’s. If it is not satisfied then it is not cannot be a self-dual function, ok. So, that is first condition, but that is not sufficient is a necessary condition not sufficient. So, another requirement is the function must not contain mutually exclusive minterm, ok. So, what do you mean by mutually exclusive minterm?

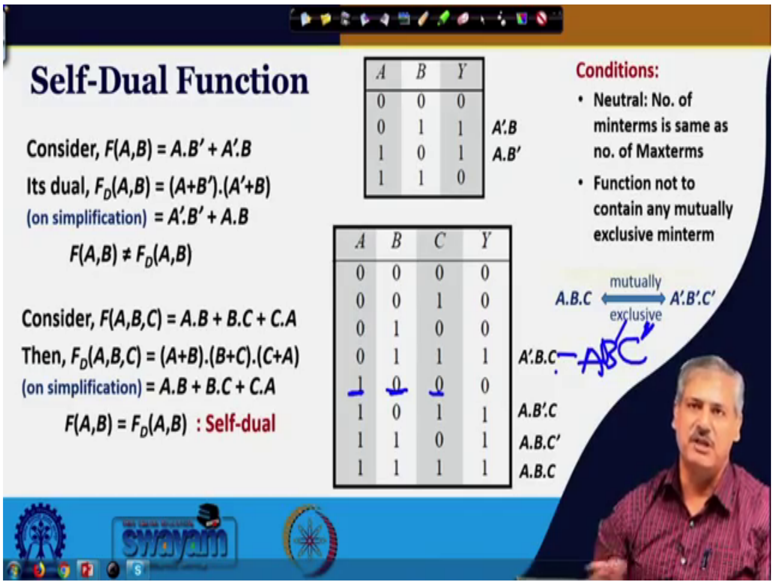
(Refer Slide Time: 33:03)



So, if there is a minterm like A, B, C, ok; its mutually exclusive minterm is A prime B prime and C prime. If there is a minterm like A prime B C it is mutually exclusive be A B prime and C prime, ok. So, this is its mutually exclusive minterm. So, they should not be part of the mutually exclusive term should not be part of the Boolean expression then it will not be self-dual.

So, in this particular is you can see that A prime B C its mutually exclusive term is A B C prime. This one, its mutually exclusive is A B C prime ok, but A B C prime is not part of is this minterm is not there. So, these A B C prime is where here.

(Refer Slide Time: 33:57)



Sorry, A; its mutually exclusive is A B prime C prime ok. So, where is A B prime C prime? So, this is A B prime C prime. So, this is 0, ok. So, that way you can see that this particular function is self-dual because if it satisfies both the cases, ok. So, anywhere it will be wherever you see some such cases if you examine if you can find self-dual function and for which if you just change the AND and OR the circuit will I mean look different, but the function output will remain the same, but without any alteration in the input-output relationship, ok.

(Refer Slide Time: 34:43)

So, with this we conclude.

(Refer Slide Time: 34:45)

And, to summarize the logical adjacency is key to Karnaugh map based representation and simplification. In SOP simplification all 1s need to be covered with the minimum number of largest integer part of 2 sized logically adjacent groups. In POS it is coverage need to be done for 0s and don't care conditions at the input side can be treated as 0s or 1s depending on whether it helps in getting a larger sized looks and then minimize the expression. And, duality at circuit level can convert from AND-OR to OR-AND and for self dual function changing and or we get the same function and does not make any difference, but from requirement of AND and OR gate may become different, ok.

Thank you.