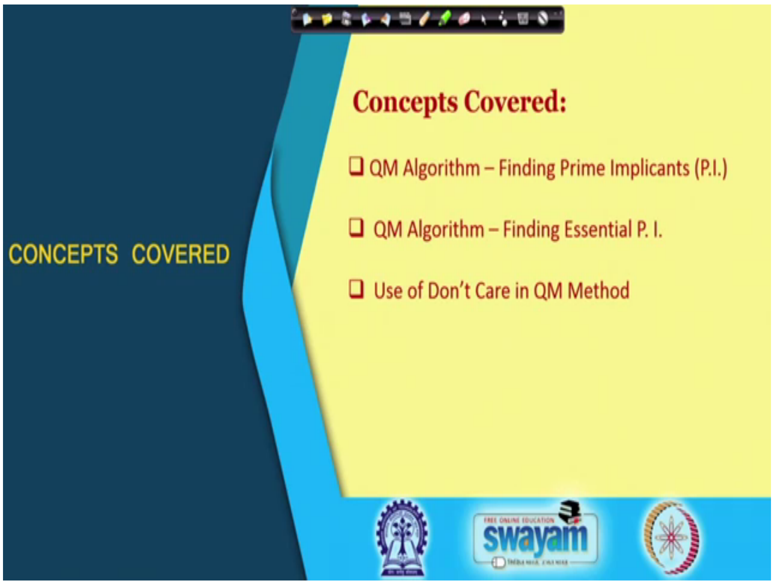
Digital Electronic Circuits
Prof. Goutam Saha
Department of E & E C Engineering
Indian Institute of Technology, Kharagpur

Lecture - 12
Quine - McClusky (QM) Algorithm

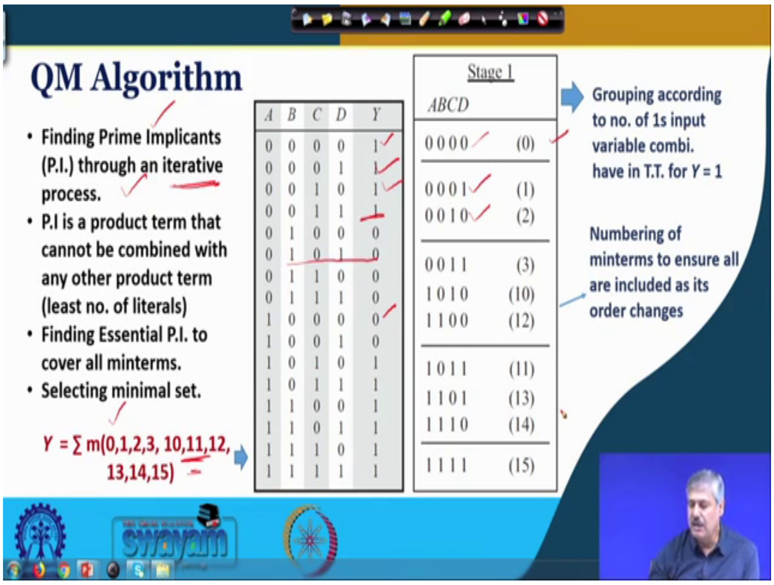
Hello everybody. In the last class, we looked at five-variable Karnaugh map based simplification, and also for also entered variable map based simplification ok.

(Refer Slide Time: 00:27)



So, in this particular class, we shall look at QM algorithm-Quine McClusky algorithm, which is also useful for simplification purpose. And we shall look into it is important steps that is finding prime implicants, finding essential prime implicants, and from that forming minimal set, and also use of do not care in QM method.

(Refer Slide Time: 00:48)



So, the QM algorithm the steps important steps are first we shall look at getting the prime implicants. Implicants are the product terms, when we say prime implicants, then these are the product terms, which cannot be combined with any other product terms. So that means prime implicants are having least number of literals for the mean terms it covers ok. So, this is cannot be further grouped with other product term.

And then from that finding essential prime implicants to cover all the min terms of a truth table, and then from this findings getting a minimal set to represent the given function. So, these are the different steps. And we take note of that finding this prime implicants, this goes through a iterative process for which a computer based approach or algorithm or programs are available or people right.

And for large number of variable minimisation of large number of variables, these corresponding functions it is very useful ok. So, to understand how it works let us look at one example the example is very simple, it is a four-variable example. So, we shall later compare with the Karnaugh map based simplification also how it works and all. So, this is having minterms 0, 1, 2, 3, 10, 11, 12, 13 and 14 ok.

So, the first step as I said is to find prime implicants. And to get prime implicants, it goes through as I said I iterative process, so let us first look at the stage-1 of the process. So, in the stage-1 of the process what we are doing, we do a grouping of the minterms in terms of input variable combinations. So, input variable combinations the first group that we are having is those minterms, where no one is present in the input combination. If such minterm exist, it will be there will be such a group form; otherwise it will not be there. So, since in this example 0 is a minterm that is present right. So, 0 0 0 0, this combination is present, so it is there ok. And there is only such one such possibility that none of none no one is present, so that is why, there is only one member in this particular group ok.

So, next we from another group with those inputs those for with for those minterms, where the inputs are having only 1 1 in their combination. So, 0 0 0 1 this is present, so this is there. 0 0 1 0, so it is there. So, the other input combination, where there is only one is 0 1 0 0 that is 4. But, 4 is not a minterm in this example, so 4 is not present so 0 1 0 0 is not present.

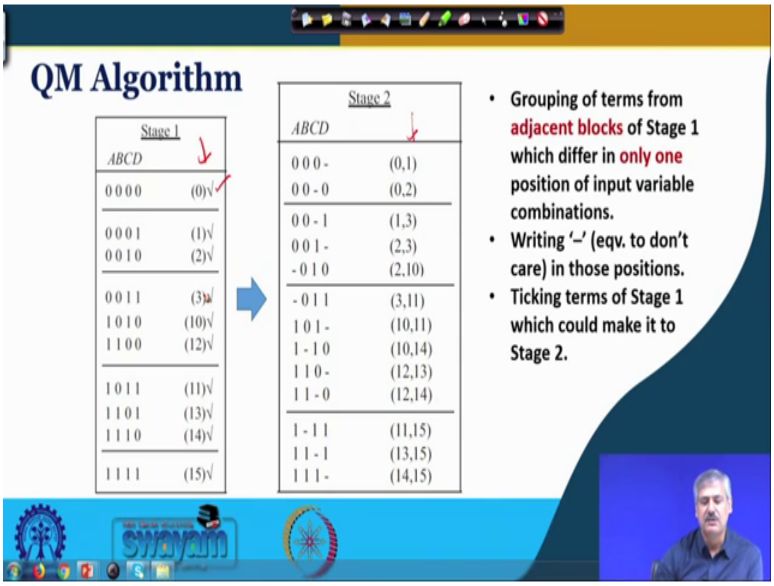
The other option is 8 ok. So, one triple 0, so that is also 0 over here that is not a minterm ok, it is not here in this particular you know example. So, it is also not member. So, wherever it is a minterm is present, then corresponding value it is may available in this particular group ok.

Now, next group we will be forming as you understand that is why I said it is a iterative process in that sense also in the way we are forming the groups. So two 1's, two 1's will be present. So, two 1's are present for the minterms, which is 3, 10 and 12. This way you can see three this is there right, other two 1's that is present here is 5 for 5 is 0, so it is not member ok. So, we have to consider the one that are present. So, 3, 10, 12 are there. So, similarly three 1's are present ok.

Next group that is 11, 13, 14; and last all four 1's are present if 15 is a member, if 15 is a minterm, then it will be there. So, it is for four-variable case. So, five-variable, six-variable case, then there will be more such groups with four-variable combinations, five with combinations of four 1's with combination of five 1's combination of six 1's and so on so forth, it will continue is it fine.

So, this is we by this we get these stage-1, the one first arrangement of this QM algorithm. And in this case the other than what has been mentioned here, since the way the minterms are appearing their order of appearance has changed. So, we just write the corresponding decimal value alongside in the right hand side to make sure that all the minterm has been properly included from here. So, no minterm should be left, so that it does not you know go out of our attention, so we are also writing it in this manner ok.

(Refer Slide Time: 06:11)



So, from that we go to stage-2. So, in the stage-2 what we do, we form groups again the way we have done before, but in a little bit different manner. So, we take adjacent blocks, why adjacent blocks we shall tell later; so adjacent blocks of stage-1. And then we see among the members there the input combinations the minterms, where I mean the just differing by one position ok. So, we form a pair of them like 0 0 0 0 and 0 0 0 1 input combination is you know just changing the D position 0 to 1, and A B C remaining same as 0 all 0, so that is only one change is there ok.

So, we take up this as a member that goes into the next stage-2 first block ok. And the place where D is changing, we put on the variable that is that position where the value is changing, so that we put as a dash. Dash means, what is the physical significance of it that whether D is 0 or 1 dash is kind of you know do not care that a minterm is produced that means, A B C is 0 0 0 D is 0 or 1 ok, there are minterms. So, two minterms are there for D is equal to 0, A B C remaining 0 all remaining 0 and d remain D being 1 A B C all remaining 0, so that is the physical significance or meaning of it ok.

So, similarly 0 and 2 we can combine, because only C position is changing. So, 0, 2 is the next member of the block, so these are the only possibilities. Now, can this first block be combined with block-3 for this particular consider the consideration that is only one position will be changing we cannot, because we know in the first block there was only no 0.

And in this particular block two 1's are there. So, they are differing at least by two positions is not it, this two positions are they are differing. So, this so because of which you just cannot, so this block this block, so subsequent blocks are differing at least by two positions not one position. So, they cannot be grouped with this particular first block ok. So, we are not considering them right.

Now, we look at whether this block and this block can be combined, and what are the different possible ways. And for that we see this one can be combined with three, because A B and D are having values 0 0 1 respectively 0 0 1 respectively. Only C is changing from 0 to 1 over here, so we write 0 0 dash 1 in this particular case. Again see this physical significance is the same that for A B and D 0 0 1 respectively. See whether 0 or 1 there are two minterms available in the truth table ok.

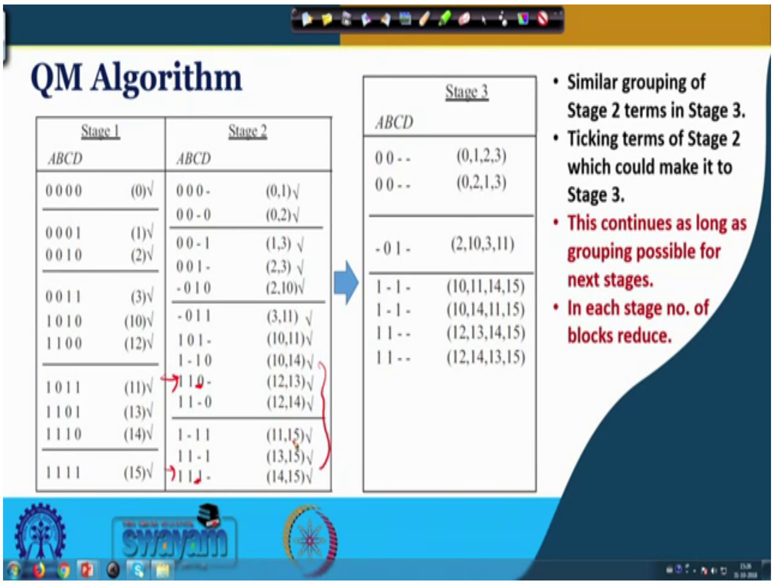
So, similarly this 2 and 3 can be combined, 2 and 10 can be combined, so this is what we can see? Can 1 and 3 be combined sorry 1 and 10 be combined, so this is 1 and this is 10 right. So, we see this is 0 and 1 they are changing 0 0, they remaining constant. Again 0 and 1 that is changing, and 1 and 0 is changing. So, more than one position it is changing is it all right. So, for which we cannot have 1 and 10 combining the way we have seen that only one position that is changing in place of that we shall put a dash rest of the values will remaining constant remain same ok.

So, next we see that this block and the other block ok, how it can be combined only as I said adjacent blocks are required. So, this 3 and 11, so 0 0 1 1 and this is 1 0 1 1 only one value is changing. So, it can be combined. So, the first value that is there which that position is only changing, so that will be a dash and rest of the values are 0 1 1 ok.

So, similarly if we keep examining all right appropriate codes, so it will be having this particular groups found ok. Similarly, for combining these two blocks in the stage-1 of stage-1, where only one position is differing, we have 11, 15; 13, 15; 14, 15 these are the combinations found, and accordingly they are called the representation is mentioned here ok.

Now, one more job is left in stage-2. So, when you do the combination, we look at where every member of stage-1 has been part of a group of stage-2 ok. If it is then we put a tick mark against it, if it is not, we keep it unticked that it could not be combined for a particular function or a truth table it might be so possible that it is not able to combine, because it is not changing in one position ok. So, it will not be ticked, so it will just remain there. So, what is its significance, we shall discuss little later.

(Refer Slide Time: 12:07)



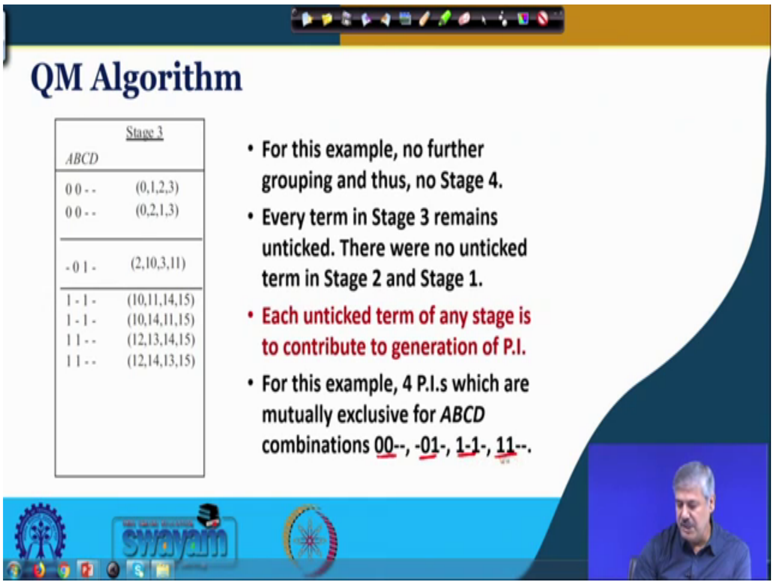
So, we have completed stage-2 as I said it was a it is a iterative process, so we move to stage-3. And what we will do in stage-3, we shall look at adjacent blocks of stage-2 right. And we see we shall see whether they can be combined following the same principle that is they are two members are differing only in one position ok. So, let us see whether it is possible or not. So, 0 1 0 0 0 dash and 1 3 0 0 1 0 0 dash 1 are they differing in one position only one position no 0 0 0 0 this is fine, 0 and dash, and dash and 1 ok, so because of which we cannot combine these two, because it is differing in two position ok.

Now, 0 0 0 dash and 0 0 1 dash, can these two be combined yes why, first two are 0 0. D is D position is dash only C position is changing in one case it is 0; in the other case, it is 1. So, this is a candidate. So, corresponding term will be 0 0 dash was already there. So, this 0 1, now a a dash is coming over here. And the term corresponding term is 0, 1, 2, 3. So, this is going as in the stage-3, is it fine? So, 0 1 is going in the stage-3, 2 3 is going in the stage-3, we can put tick mark against them right. Next 0, 2 and 1, 3 can be combined right.

Similarly, if we look at examine 0, 1 and 2, 10 whether they it can be combined or not, we see that it is differing in many places. So, 0 dash is differing 0 0 is not differing, 0 1 is differing, and dash 0 is differing. So, three places it is differing. So, they cannot be combined ok. So, once this part is done between these two adjacent blocks, we shall look at these two blocks. And between these two blocks, we can see that 2, 10, 3, 11 2, 10 3, 11 they can be combined. And rest of the combinations are not possible, because it is not it is differing more than one places ok. And 2, 10 3, 11 dash 0 1 0, and dash 0 1 1. So, 0 1 is and dash is same 0 and 1 this is only changing, so that is a dash over here which is coming here ok.

And finally, between these two blocks ok, so these are the different possibilities if we just examine the same way, we shall get it the 10, 11, 14, 15 and 12, 13, 14, 15 they are there and the corresponding terms are this ok. So, 12, 13, 14, 15; if we just take an example say here and 14; so 1 1 0 dash and 1 1 1 dash. So, 0 and 1 that is only differing this position and this position, so we put a dash ok. So, we can examine one member against the other, and we can do it, and if we writing a code, then we can write a code for that ok. See in doing that we see that 0, 1; 2, 3; 2, 10; 3, 11; 14, 15; 12, 13 all the members have found a place in the next blocks. So, each one of them is ticked is it ok.

(Refer Slide Time: 16:03)



Now, finally when we are we have reached stage-3, we shall look at what we shall look at whether we can move to stage-4 by combining this right. So, if that is the where they differ only in one place is there any such case, so if you look at 0, 1, 2, 3, and 2, 10, 3, 11, so if you compare these two ok, so 0 and dash it is differing 0 0 not differing, dash and 1 it is differing, dash dash it is not differing.

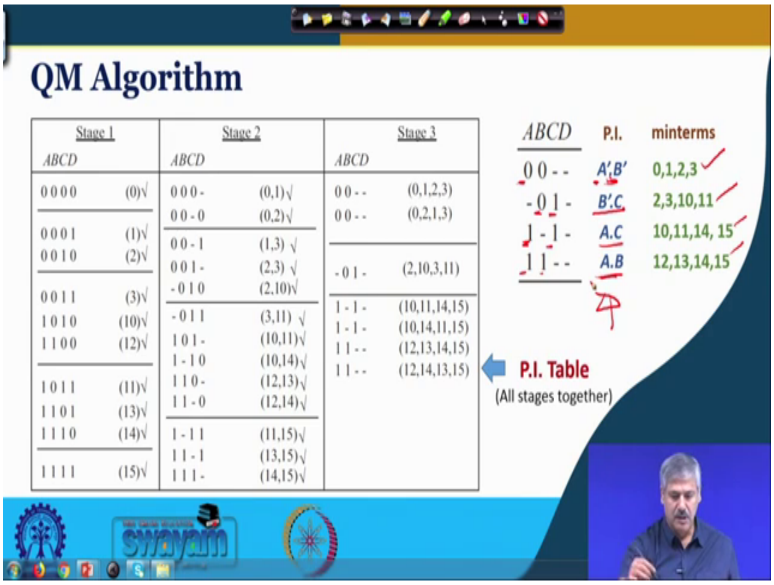
So, two places it is differing zero dash this position A position and C position. So, they cannot be combined ok. So, either 0, 2, 1, 3 and 2, 10, 3, 11 can be combined. If we examine this and this ok, 0 dash A position it is differing, B position not differing, C position it is differing, D position not differing ok. So, it is differing here, and it is differing here. Again two positions it is differing. So, it cannot be further combined only one position, it should change to become a member of a group in the next stage.

So, similarly over here if you look at 2, 10, 3, 11 and 10, 11, 14, 15 whether they can be combined just for example, so this is different A position is different, B position is different, C and D are same ok, but A and B are different, so because of two positions differing again they cannot be further combined. So, we can combine examine each one of them. And see that none of this can go to the next level, and because of which there is no stage-4, and also none of this member over here is ticked ok.

Now, each unticked term, now we understand the significance of this unticked term. So, each unticked term is a member to contribute to prime implicant generation. So, each unticked term will generate a prime implicant ok. So, here all these will be coming from this stage, but if any previous stage there is a unticked term ok, so that we has to be brought forward in the final prime implicant the that has that we get out of this exercise ok.

So, in this example so these are the four prime mutually exclusive, because you see 10, 11, 14, 15 and 10, 14, 11, 15 they are the same they remain the same, so that is 1 dash 1 dash. So, they are not essentially different. So, basically these are the four terms that we get, which will contribute to the prime implicant formation ok.

(Refer Slide Time: 18:51)



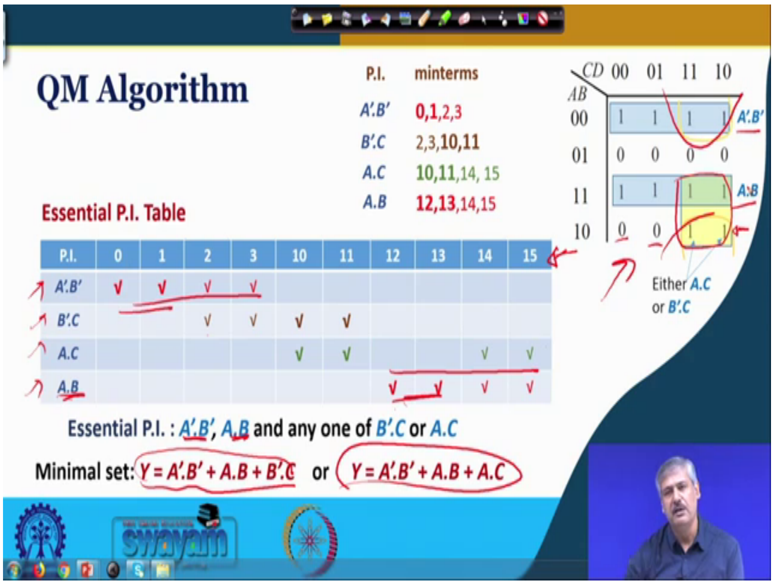
So, all the three stages combined together, it would look something like this the one that you see right earlier we had shown it at a separate manner. But, when you work it out, you can just keep on you know adding tables and columns and all till it converges. The convergence is guaranteed, because you can see in each of this cases the number of this blocks the groups are coming down from stage-1 to stage-2, stage-3, it was you know 1, 2, 3, 4, 5 here, 4 here, 3 here right, so that way the one I mean the extreme scenario will be that all are dash A, B, C, D, E all are dash that means, it does not matter, the output is 1.

So, Y is equal to 1 ok, so that is one extreme case that you can see that irrespective of any change in the value. So, all input becomes kind of you know do not care any value whether it is 0 1 any combination output is 1 ok, so that is the you know kind of meaning one can bring out for such cases, but it converges I mean it cannot go beyond that. Otherwise, the convergence we had seen the where that after stage-3, it cannot be further grouped. So, it does not go to stage-4 ok.

So, this is how it will (Refer Time: 20:04). So, this is the PI table this is called the PI table, and the corresponding terms that we get this 0 0 dash dash and the terms. So, the way we define it wherever there is a 0, we write is as A prime corresponds to A, and corresponds to B, so 0, so it is B prime ok. So, the corresponding product term representing this prime implicant is A prime B prime. And the minterms that is covered by it is 0, 1, 2, 3.

So, B C B is remaining constant with 0, and C is remaining constant with 1. So, B prime C is the corresponding prime implicant. A is remaining constant with 1, and C is remaining constant with 1. So, A C is also the A C is the corresponding prime implicant. Finally, 1 1 a for A and B, so this is A B is the prime implicant. The minterms that are covered, we can see at this side by side it is mentioned ok, this is understood how we are getting. So, the first step is getting prime implicants, and through a prime implicant through an iterative process the way we have described so for and for these example we arrive at this.

(Refer Slide Time: 21:23)



Now, we have got the prime implicants, we have got the product terms, if we just sum them up algebraically, we will get Y which is correct, I mean there is nothing but whether it is giving way minimal set or not, it is not yet ascertained ok. To ascertain that we shall look at next stage which is next step which is called as finding essential prime implicants.

What does it do? In essential prime implicant table, we are looking at the requirement of the coverage of all the minterms by the prime implicants that we have got in the previous step through the prime implicant table. So, the minterms that need to be covered for this specific example as 0, 1, 2, 3 0, 1, 2, 3, 10, 11, 12, 14 that is the one that you can see right.

And then we write the corresponding prime implicants that we have got, and what are the minterms that are being covered by them. So, the coverage by A prime B prime is 0, 1, 2, 3; B prime C is 2, 3, 10, 11; A C is 10, 11, 14, 15; and A B is 12, 13, 14, 15 is it fine. So, this is the way we get the prime implicant table right. So, this corresponding tick marks are there.

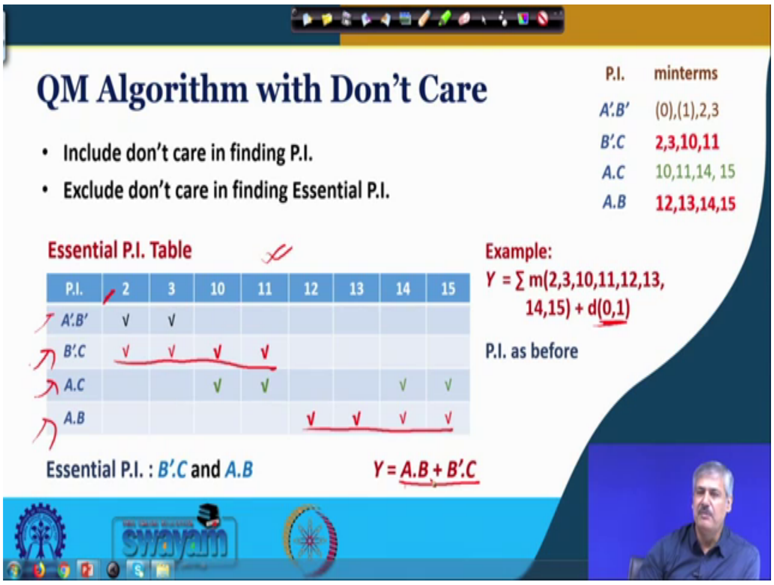
Now, way we look at this essential prime implicant table, and the corresponding you know these prime implicant prime is A essential prime implicant. So, A prime B prime is a essential prime implicant is an essential prime implicant understand this point, because if it is not considered, then 0 and 1 minterms will not be covered. The same way 12 and 13 is covered only by A B and not by any other prime implicants. So, A B is also an essential prime implicant, so this is there.

Now, when A prime B prime, and A B and A B, these are included, these are this need to be considered in the SOP form right. So, what is the minterms that remain yet to be covered, they are 10 and 11 they are 10 and 11, because A prime B prime, and AB are covering 0, 1, 2, 3 12, 13, 14, 15; so 10 to cover 10, 11. We have got two terms with you know same similar complexity, if we consider that the inputs are available in complement and uncomplemented form, so one is B prime C, another is A C ok.

So, one is covering 10, 11 as well as 2, 3; another is covering 10, 11 as well as 14, 15, so we can take either of them ok. And we can get one set of you know essential prime implicant using this a minimal set like this, another minimal set like this ok. So, this is the two possibilities, these are the two solutions that we can get out I mean two possible solutions with similar complexity from the QM algorithm for the given example.

And we can examine whether this is or not by the K-map simplification that we are already familiar with. So, the minterms if we map it in the K-map, this is how it would look like ok. And one group A prime B prime comes over here, A B comes over there ok. And the remaining two 1’s that need to be covered, this is 10, 11 in the Karnaugh map, you know this is 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, so this is the 10 11. So, 10 11 can be grouped to like this particular one or this one the way it has been seen ok, two possibilities are there. So, in one case will get A C, another case it will get P prime C. So, one of them need to be considered, so that 10, 11 remain covers remain covered, and A prime B prime, and A B must necessarily be present ok.

(Refer Slide Time: 25:52)



So, how we find the QM algorithm based simplified expression, if there is a do not care case, so do not care means in at the input side we do not care. So, we know this x by which we are presented them in the Karnaugh map. So, what we do the method is simple, these do not care we considered as minterm in the generation of prime prime implicants ok.

So, when you do that there is a possibility of getting largest sized group ok, and we exclude it, when the coverage is required in the finding of minterm coverage is required in the finding of essential prime implicant, this is what we do ok, and by which we can ensure a minimised or simplified expression. So, we can take this example, which is just a variation of the example that we have taken. So, 0 1 is the do not care, earlier we considered minterm as 0, 1, 2, 3 up to 14, 15. Now, we are considering minterms are 2, 3, 2, 15, and 0, 1 are the do not care.

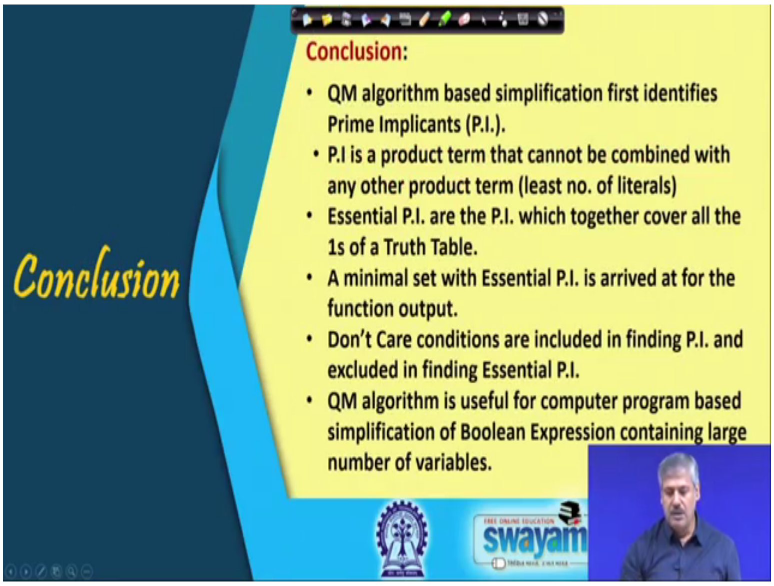
So, in the first stage as I said 0, 1 will be considered as minterm to find the prime implicants. So, the prime implicants that we had got in the previous example will be this one in this particular example as well. So, the prime implicants we shall get are this A prime B prime, B prime C, A C, and A B by the same method.

Now, in this example, because 0 1 is something which need not be covered ok, it is do not care. So, 0 1 is excluded, you see 0 1 is not over here, we are starting from 2, 3 2, 3, 10 to 14 ok, so accordingly it has been written. And what we see in this essential prime implicant formation that B prime C, and A B between themselves 2, 3, 10, 11 and 12, 13, 14, 15 all the minterms are getting covered ok.

So, the minimised expression, we get is A B plus B prime C. So, this is what we do in case of QM algorithm based this method ok. And we are not discussing is similar thing for POS, but we have already noted that from the dual circuit thing. So, we can find A prime for say this example, so we can get we can get F prime, where this will be one, and this will be one, and rest of the things will be 0 by which we can get the expression ok.

And then we take the compliment the dual of it, dual of it means AND becomes OR, OR becomes AND, and the input, outputs are complemented. So, F prime we have considered, so output becomes F F ok, and inputs will just get complimented, and then AND becomes AND, OR becomes AND means, we will get the POS representation ok.

(Refer Slide Time: 29:14)



So, to conclude QM algorithm based simplification first identifies prime implicants. These are the products terms, which can may not be combined with any other product terms, so it represents a least number of literals. And essential prime implicants are the PI, which together cover all the 1s of the truth table. And minimal set with essential prime implicant is arrived at for the function or getting the function final function output. Do not care conditions are included in finding prime implicants, but excluded in finding essential prime implicants for in completely specific truth tables having do not care. And this is useful for computer based simplification of a Boolean expression.

Thank you.