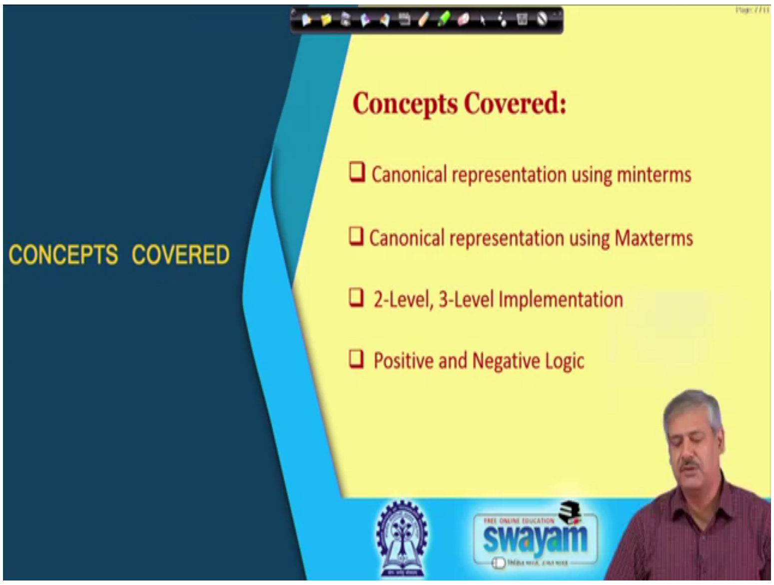
Digital and Electronic Circuits
Prof. Goutham Saha
Department of E &EC Engineering
Indian Institute of Technology, Kharagpur

Lecture – 09
Truth Table to Boolean Function and Implementation Issues

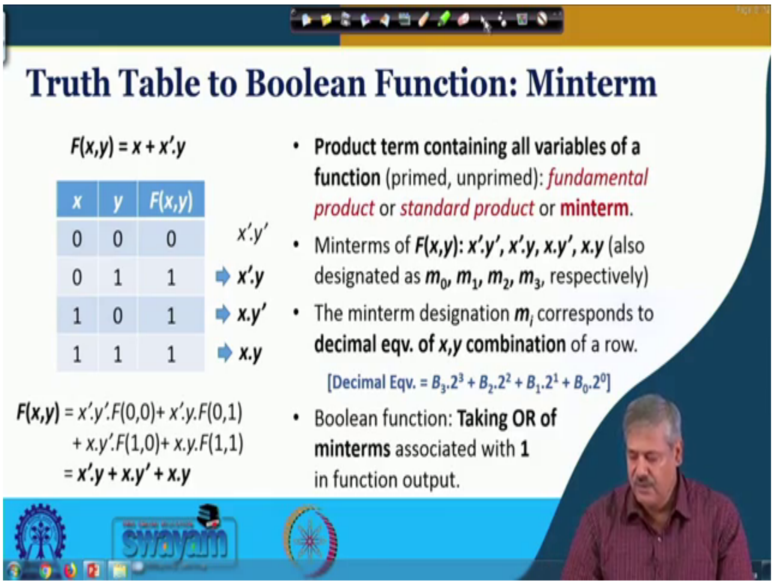
Hello everybody, in the last class, we had seen if a Boolean function is given how to convert it to a how to get a truth table out of it, and we have also saw how to implement a Boolean function, and how Boolean algebra can be used to simplify a Boolean expression and get a minimized expression out of it. And we also discussed the Shanon's expansions theorems.

(Refer Slide Time: 00:41)



And in today's class we shall discuss the canonical representation of a truth table using minterms and also is max using maxterms. And we shall discuss given a canonical representation how we can get a two level implementation of it. And also in side by side, we shall discuss multilevel implementation. And we shall end with positive and negative logic.

(Refer Slide Time: 01:07)



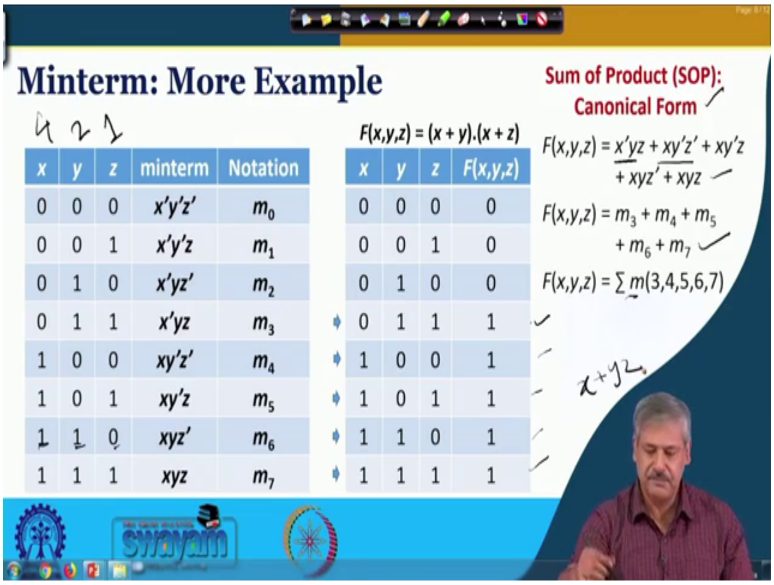
So, in the last class, we ended with the nested version of Shanon's expansion theorem. And we arrived at for a two variable problem. And expression if x plus x, x plus y you just took this was an example. So, you can may not look at it this point of time. So, this is the truth table. And this particular truth table we can see that there are 4 such terms. And the terms that are associated with 1 from the Shanon's expansion theorem if we just compare these are the terms.

And this was there in final expression that we had formed. So, these are called product terms and which is finally summed ok, the end terms which is finally OR ok. So, interesting thing is here that only the one that are having function output 1, which is considered in this summation process or adding process. The one that is having 0 is excluded ok. And the another interesting thing is if you look at the x y combination, so when x and y both at 0 right, so the corresponding term is x prime and y prime ok.

And 0 1 if x is equal to 0 this is x prime and y this is just y un complimented version that we find. So, x is one and y is 0 this is x and y prime and both of them are one both are un complimented. So, these are the corresponding term that we get which is called also minterm in this two variable problem. So, this minterm or fundamental product or some standard product is nothing but a product term continues all the variables ok. So, this x over here is not a minterm right, but x prime y this is a minterm because it contains all the variables. It could be in complemented version or un complimented version primed or unprimed ok, it does not matter, but these are there.

So, you can see that for each of these two variable problem, 4 possible such minterms are there. This x prime y prime, x y prime and x y. And what we do? We just for convenience we term them also using m 0, m 1, m 2, m 3, this suffix 0 1 2 3 comes from there decimal equivalent of this x y values. So, 0 0 the decimal equivalent to you know is 0, 0 1 is 1, 1 0 is 1, and 1 1 is 1 0 is 2, and 1 1 is 3, so that is why m 0, m 1, m 2, m 3 that is how it is mentioned. And finally, the function is obtained by taking OR of all the minterms that has got 1 in the function output that we have that we notice here ok.

(Refer Slide Time: 04:21)



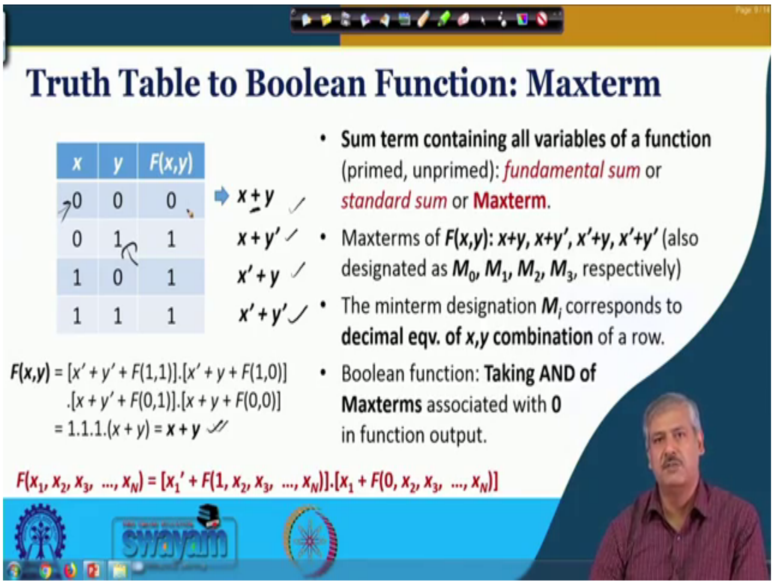
So, that was a two variable example. For three variable, what we will do? So, we shall have the corresponding minterms. The minterms will be having all the three variables and the possibilities are 8 with three variables. So, 0 0 0 ok, for this combination at the input. So, the minterm will be x prime, y prime, z prime all of them wherever 0 is there, the corresponding time term will be prime. 0 0 1, it is x prime, y prime, z because z is equal to 2, so z is unprimed or un complimented, so that way we will go up to when all of them are 1, it is x y z. And the corresponding a notation you in the in terms of m 0, m 1, m 2 go up to m 7 which is the decimal equivalent of this place ok.

So, we already have noted that the binary equivalent of 1 1 1 is this place value is 4 ok, this place value is 2, this place value is one. So, if it is multiplied with 1, 2 and 4 here each of these cases, so 4 plus 2 plus 1 is your 7. In this case 4 plus 2, so this is 4, this is plus 2, and this is 0 multiplied with 1. So, it is 6. So, this is m 6 ok. So, this is the way we get the decimal equivalent and the corresponding notation is presented here ok.

And if you look at this particular truth table, this truth table we have seen the before, but here that is why we are looking at given a truth table how we are converting it to corresponding Boolean expression. Here we are talking about expression that has got minterms. So, we shall look at the terms that are having all ones. So, these are the terms that are having ones 1, 2, 3, 4, 5. So, 5 minterms will be there and which will be summed to get the corresponding Boolean expression right. So, this is the one where the first term x is 0, y and z are 1 for which the minterm is generated. So, this is x prime because it is 0, and y z, so x prime y z that you see.

Next one is x is equal to 1. So, x it will be x then y is 0 and z is 0. So, y prime z prime, so x y prime z prime. So, this way you will get all the five terms. And this five terms you can see m 3, m 4, m 5, m 6 and m 7 that is what you do here. And you can use a for a more compact representation using a sigma a summations sign where m stands for minterm the small m and 3, 4, 5, 6, 7. So, these are the corresponding decimal equivalent places that is what you finally, sum it up. And this is called canonical form right. And you can have a normal sop form also which is just a we said the x plus y z that is not called canonical form when all the variables are present in you are coming up the minterm, it is called canonical form representation. So, this was a sum of product kind of you know representation of the truth table.

(Refer Slide Time: 07:35)



And you can have the product of sum the dual version of it. So, for that the term that was that are used are called maxterm ok. So, in how this maxterms are formed and designated. So, again if you go by the dual of the Shanon's expansion theorem the one that we had ok, so if you expand the function that we had earlier this one that we just used.

So, if you for this particular truth table if you keep expanding it x plus x prime y right, and then you know all the individual terms, and you substitute it over here, only we know that F 0 0 was 0, and rest of the terms was 1 by calculation. And then when you do it all individual terms you generate 1, and only the last time will generate 0. So, 0 plus x plus y and rest of the terms are only 1 ok, so that gives you x plus y. So, this is you are getting from nested version Shanon's expansion theorem these dual version ok, so that you can see.

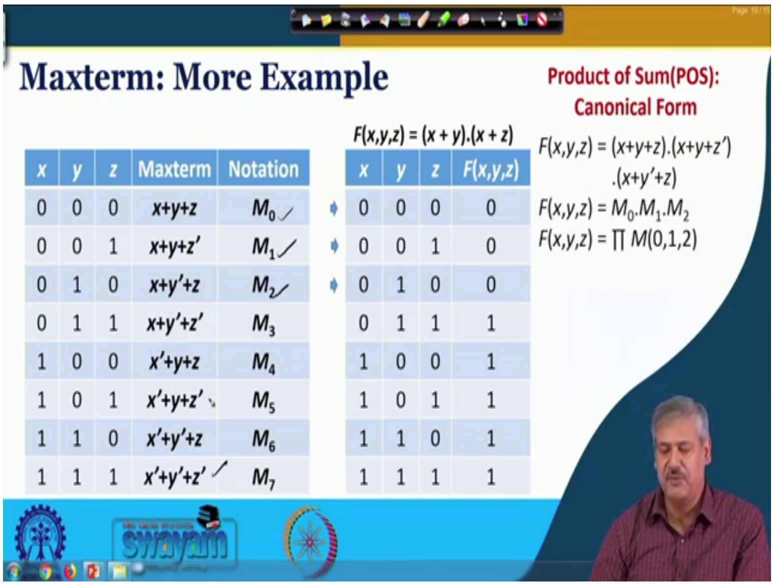
And the corresponding the truth table this corresponds to only one location that is 0 and for the maxterm definition wherever there is a 0 we take is we take it as un complimented unprimed. And whenever there is a 1, we take it as a complimentary. So, this is the difference important difference we have to take note of between minterm and maxterm designation. And of course, this is a sum term. So, there is a OR sign in between two variables; in the other case there was a AND sign product sign between two variables. So, this is the important difference.

So, we see that the for two variable these are the 4 possible cases where all the literals, all the variables are there and the terms are 0 0 - x plus y, 0 1 - x plus y prime, 1 0 - x prime plus y, because it is 1, it is x prime; x is 1 that is so why it is x prime y is 0 that is why it is y ok, 1 1 this is x prime plus y prime. If it is 1, as I said it could be a prime; and if it is 0, it will be unprimed ok.

So, coming to the definition, so some terms containing all variables of a function which is there in a particular in variable problem, so that is called fundamental sum or standard sum or maxterm ok. And similar to minterms we also designated with m 0, m 1, m 2 and m 3, where 0, 1, 2, 3 corresponding you know decimal equivalents. So, 0 0 this is M 0, this is M 1 and so on and so forth ok. This capital M represents maxterm, small m we have used for minterm.

And a final expression, we get for this particular truth table just by we are taking product of taking AND of all the maxterms that you have got for a particular function, where it the function output containing 0 generates that particular combination of the input generates a specific maxterm clear.

(Refer Slide Time: 11:19)

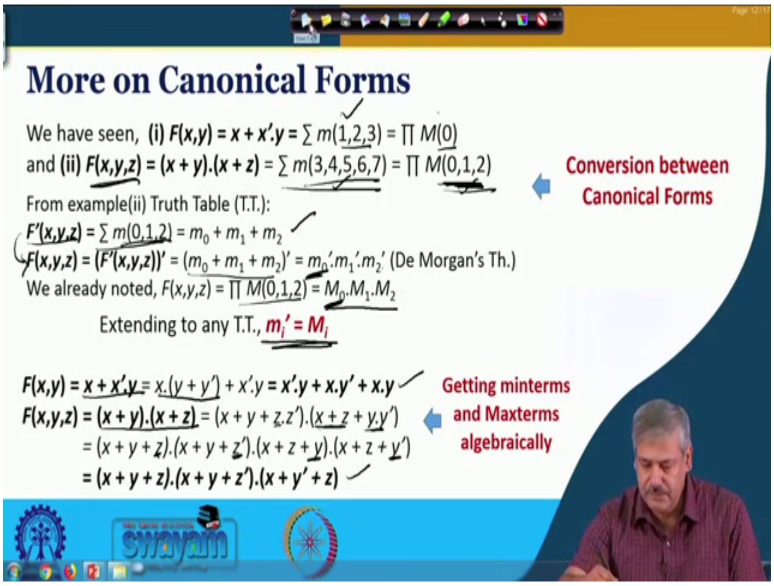


So, we look at it for the three variable example ok. So, this x y z, all of them 0. So, if it is 0, then it will be unprimed un complimented. So, the corresponding maxterm is x plus y plus z ok. 0 0 1, it is x y, and z is 1, so that y z prime will be there, so x plus y plus z prime. And that way when it is 1 1 1, we have got x prime plus y prime plus z prime. So, this is three variable you can extend it to n variable and the location wise so this x 0 0 0 is decimal equivalent is 0 0 0 1 decimal equivalent is 1, 0 1 0 decimal equivalent is 2. So, you have got maxterms M 0, M 1, M 2 up to M 7 ok.

For the same truth table that we had seen before if you remember. The same truth table we had five 1s here right, and the mean terms contained 3, 4, 5, 6, 7. Now, we are trying to represent it using maxterm. So, the maxterms that are there which need to be considered or the place where there is there are 0s ok. So, three 0s are there three maxterm will be there which will be finally ended. So, we have to see that how these each method maxterm is represented. So, 0 0 0 these are the three different values of x y z for the first maxterm over here M 0.

So, this is nothing but your x plus y plus z, x plus y plus z, so that is that goes here. So, next one is x plus y plus z prime, and third one is x plus y prime plus z, is it ok. So, this is your M 0, M 1 and M 2. And more compact representation this is the product phi that is that means, that multiplication of maxterms 0, 1, 2 that is ending of 0 1 and 2, three maxterms ok. Is it clear? Ok, so, this is also called canonical representation because all the variables all the maxterms are involved.

(Refer Slide Time: 13:49)



Now, given one particular form can I get the other form out of it? So, the examples we have seen in the first example we had seen that three ones where there and 1 0 was there. And for minterm and the minterm representation SOP this is what we got ok. And for maxterm rep representation only one term was there other, so that is why you can just write it you can remove the phi that is M 0. So, if you see that there are 4 possible terms in the terms you know decimal equivalent 0, 1, 2, 3. So, if 3 goes here, the remaining one out of the whole set is there in the case. So, if minterm contains 1, 2, 3; maxterm will contains 0.

In the next example, we have seen the minterm contains 3, 4, 5, 6, 7; and maxterm contains 0, 1, 2. So, for three variable problem possibilities are 0 to 7, eight possibilities. So, if 5 is taken there in minterms there is representation SOP. So, rest three is remaining three is taking up in the maxterm base representation using POS form - product of sum form. So, the whole set minus 1 particular from representation whatever are the corresponding decimal equivalent places, then remaining is going for the other forms, so that is the way you can quickly convert from one form to another.

The other important thing that we can observe from whatever discussion we had so far is in the first example, this sorry second example the F x, y, z. So, the minterms are 3 4 5, 6, 7. So, if you take complement of x f x y z whatever we had ok. So, wherever 0 was there, 1 will present; and wherever 1 was there, 0 will present. So, F prime x, y, z that is complement of x F x, y, z the one that we had will be having minterms 0, 1, 2.

So, this in the SOP form summation of this three minterms that that would be giving you the complement of F x, y, z, the y had defined in the truth table. So, you can write it in this way m 0 plus m 1 plus m 2. So, how can we get F x, y, z from a F prime x, y, z, you just a complement of it right. If you take complement of F prime x, y, z that is double inversion is the same you know the function itself. So, that is what is that m 0 plus m 1 plus m 2 taking complement of that ok. So, by De Morgan's theorem, this is m 0 prime ended with m 1 prime with ended with m 2 prime ok.

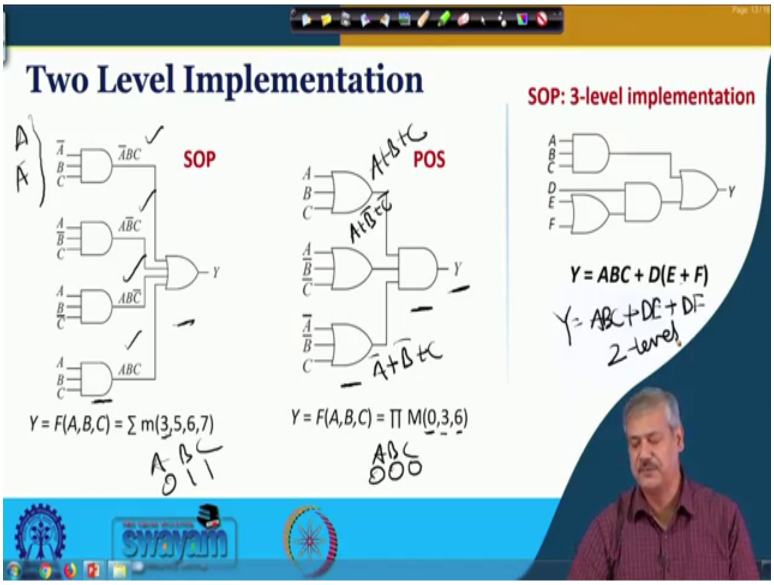
And we have already noted for this maxterm base representation the same function F x, y, z, F x, y, z over here, we have noted that this is M 0 capital M 0, capital M 1, capital M 2. So, we can see for any such truth table, if you just extend it that m i prime mi prime. So, M 0 prime m, M they correspond. So, m i prime is same as a minterm ith position. If you take complement of it you get maxterm the corresponding maxterm in the i th position. This is very simple relationship we will keep in mind ok.

Now, given an expression to get the minterms or the maxterms right, what we have done. So, we get we are getting the truth table. And from the truth table we are looking at the x, y, z, or xy whatever the input variable combinations. And wherever there are ones we are getting minterms for those combinations and writing it in a particular manner and similarly for maxterm. So, this is one way we can get all the minterms and maxterms of a particular Boolean expression. It can be obtained algebraically also without going via the truth table root.

So, for example, this x plus x prime y is there ok. And we want to get its this particular terms minterms, that means, all the different things that are there. So, x you can write as y plus y prime is the from complimentary this x is ended with one. So, 1 can be written as 1 y plus y prime. Then if you just expand it that is and x y with x with y and x with y prime, then you get this thing. So, this is the all the contains all the literal all the variables, so this three minterms that you get ok. So, you do not need to go in that direction.

So, similarly for the other problem POS problem. So, x plus y, you can write as x plus y z z prime complementarity it is a summing up with 0 ORing with 0, it does not make any difference in the final outcomes. So, x plus z, you can write as y plus y prime. Then you can use distributive law. So, these x plus y z will be coming here and z prime will be coming here. And in the other case x plus z is one thing and y y prime. So, y will be coming here and y prime will be coming in the place. Then you can say x plus y plus z are there in three places, x plus y plus z are there in three places sorry in two places keep one of them and these are three maxterms that you are getting that is M 0, 1, 2 ok. So, algebraically also you can arrive at the maxterms ok.

(Refer Slide Time: 20:03)



Now, implementations of these canonical form is very standard. So, SOP form. So, we just look at us to one example. So, in this case, we will be having AND bank followed by OR gate. So, this is example where m minterms 3 5, 6, 7 ok. So, they are getting ORed ok, in the final summed up in the final function output.

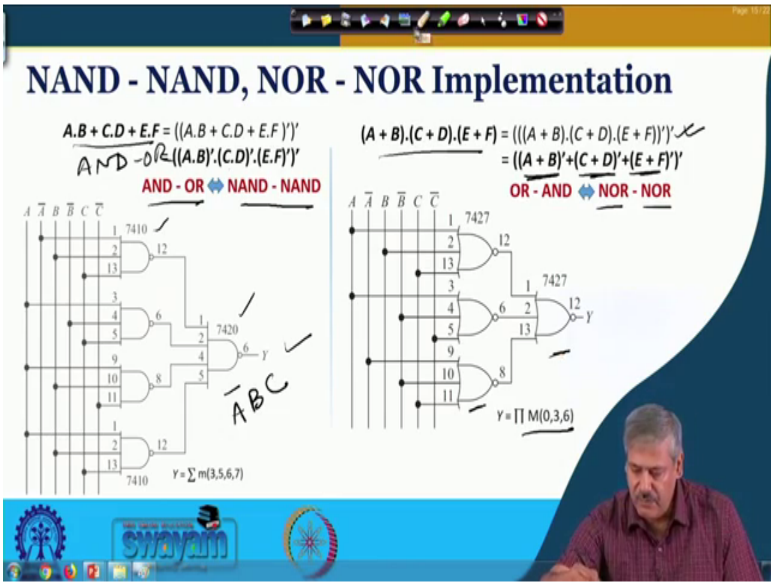
So, m 3 is A bar B C right, A B C 3 is 0 1 1 ok. So, A bar B C is one input generating one minterm. 5 is generated in the by this; 6 is generated by A B C prime; and 7 is generated by A B C ok. And finally, all of them are ORed, and you get a SOP realization of this. This is called two level ok assuming that the variables are available both in compliment and uncomplimented form.

So, both A and A bar, they are available at the input side; otherwise if it is only A is available to get A bar you need a another NOR gate ok. So, in that sense, one may think that we will increase another level ok, but the assumption over here is that both A and A bar are available simultaneously at the input side. And these are the two levels one AND bank followed by a OR gate which is required for this two level imply implementation of the canonical form of SOP ok.

For POS, it is similar. So, they will be or bank followed by a multi input and gate depending upon how many OR gates are there ok. So, practically this is as I have mentioned before the fan in, fan out all those things need to be considered. If it is violated we have to take other measure by which you see that things are in order. So, here it is 0, 3, 6 right 0 means A plus B plus C. So, this is your A plus B plus C ok. So, 3, 3 is your A plus B bar plus C bar right, is this ok. So, 0 as we have noted 0 0 0 so whenever 0 is there it will go out un complimented for maxterm generation. So, finally, 6 is A bar plus B bar plus C that is your C. So, all of them these outputs are summed sorry ANDed, and we get the final output. So, this is also two level implementation.

And what could be multilevel implementation. So, this head is just an example of a three level implementation right. So, A B C D is ANDed with E plus E and F the summation of the E and F. So, E and F are summed up then this is there. So, you can see three stages are there right. So, this is a three level implementation. It could have made it two level implementation by expanding D, E, and I mean just ending it following the distributed law and D F. So, this would have given you a two level, this would have given you a two level implementation ok.

(Refer Slide Time: 23:39)



Ah. So, you can use NAND-NAND and NOR-NOR based two level circuit for realization of a SOP and POS form respectively. This is useful in certain sense that it uses only one variety of gates ok. And this is evident from this particular relationship what has been shown here for say AB, CD and EF this is a SOP realization. So, in this case you can take double complement of it. And then from De Morgan's theorem you can see that this AB becomes AB prime that is your NAND operation over here and this is NAND operation of a CD NAND operation of E F.

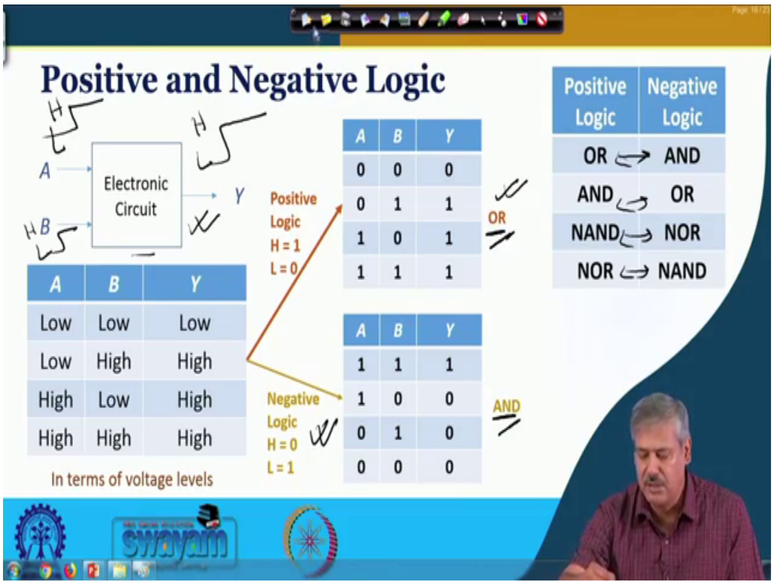
And ultimately all of them each individual NAND output are ended and complemented that means, another NAND operation is involved in the last stage the second stage second level, and you get the same SOP instead of AND and OR that is this you are having a NAND replacing the AND bank and another NAND replacing the OR gate the final OR gate ok. So, by this you are getting only one variety using only one variety of gates.

So, here is an example. So, here you see the m 3 this is the minterm that are getting something using the canonical form. It can be used for other form as well any form like what you had shown as an example. So, basically minterms 3 is your A bar B and C. So, you see this output here it is A bar B and C, so that is the corresponding NAND term will be generated and then that minterm is finally, again with the other NAND get outputs, another NAND get is formed and then you get the same SOP realization, but using NAND NAND implementation ok. And here in these example we are showing the corresponding IC numbers 3 input, 4 input IC numbers, this 7 4 1 0, 7 4 2 0. And if you remember for two input NAND gate it is 7 4 0 0 so that you have seen before.

The same thing goes for POS representation using NOR-NOR gates. And in this case again the similarly the original expressions say AB plus CD plus EF any such example you can take up. So, this is or gates or banks in the beginning which is ended in the final stage second level ok. In that case you again you are taking double complement. And then you are taking using De Morgan’s theorem.

So, this is one NOR gate; this is another NOR gate; this is another NOR gate, finally, a three input NOR gate combines all of them and you get a NOR-NOR level representation. So, similarly for this particular example, we have one NOR bank followed by another final NOR output stage, and we can get the same result ok.

(Refer Slide Time: 27:01)



So, to end we just take note of one interesting aspect of the circuit level realization. So, actually electronic circuit that we are we are we are using it is just a you know it is an entity which is just taking up some input voltage and generating some output voltage. So, for example, an electronic circuit which takes up A and B as a input, input two it is a digital electronics switching is involved.

So, input is only logic only logic only voltage level or high or voltage level low. So, these are the two cases that are there. So, basically this high or low that is you are sending. So, this is low and this is high ok. And the corresponding output is also generated which is either low or this is high. So, this is also similar the possibilities right. And the circuit that we are having has got these characteristics.

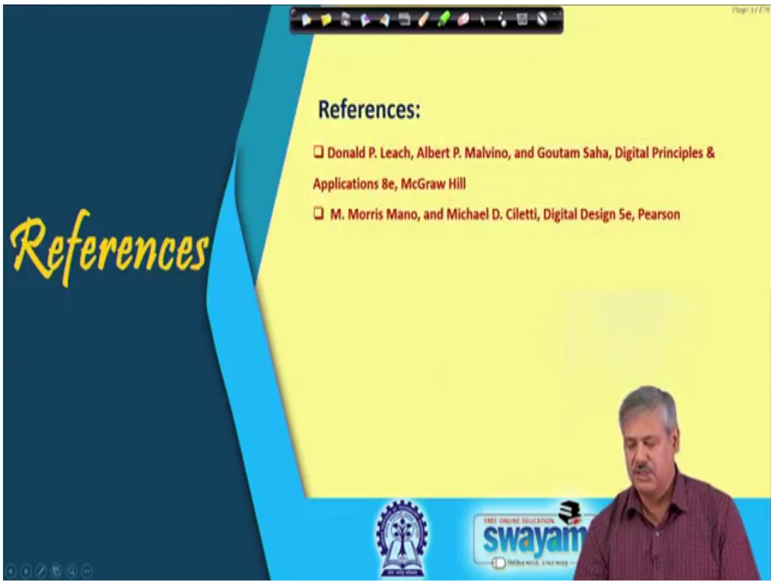
So, if input is low and low, the output voltage is also low; input is low and high output voltage is high; high and low, this output voltage is high; and high and high, this output voltage is high. So, if we consider high voltage as 1, binary value 1 and low voltage as binary value is 0 ok, then we can get this can be converted to a table like this 0 0, low; low 0 0, output is 0; low high 0 1 - output is 1, and so on and so forth ok. And corresponding the truth table that we get is that of OR operation OR logic operation, is not it.

So, when high is associated with 1, high voltage is associated with 1, and low voltage associated with 0, we say that what we are following is called is positive logic. In our discussion before now and in future also, we will primarily deal with positive logic ok, but we need to take note of negative logic. And we will use it as it occasion arises ok. So, in the negative logic, what happens it is just opposite of it. So, a high voltage is treated as logic low, and a low voltage high voltage is treated as binary value 0 or logic falls and low voltage is treated as binary value 1 or 2 ok.

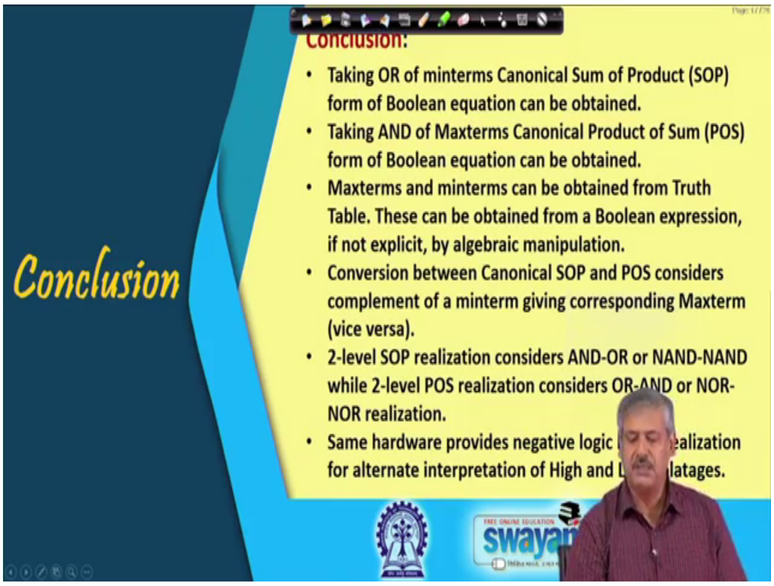
So, in that case the low, low becomes 1 1 ok, because low is 1 and low this is 1. So, low high is 1 0. And the output is high means here high voltage means binary value is 0, so this is 0 ok, so then 0 1 0 0 0. So, if you just look at this table ok, we can see that it matches with and truth table only thing that 1 1 is written in the beginning, it does not matter truth table all possible combinations we are talking about whether you write 0 0 in the top or 1 1 in the top the meaning and significance remaining remains the same.

So, same electronic circuits same electronic circuit the box that you see here which generates a high or low voltage depending on the low and high voltage is presented at the input, we get the corresponding these values ok. So, the same circuit is positive OR gate and the negative AND gate ok. The same way for any other circuit you can see there is a inter connection the same circuit we can which you can treat as a positive OR, NAND-NAND and NOR gate following negative logic this notation this understanding we can have equivalent AND, OR, NOR and NAND gates. So, basically this is the thing that you can get ok. So, this we shall keep in mind and if we required you can make use of it in our analysis.

(Refer Slide Time: 31:03)



(Refer Slide Time: 31:05)



So, with this so we conclude that taking OR of minterms canonical SOP form can be obtained. And taking AND of it you can get POS form and the maxterm minterms can be obtained from truth table as well as by algebraic manipulation of Boolean expression. And the canonical form can be converted from one from to another. And 2-level SOP realization can be done through AND-OR as well as NAND-NAND. And pos realization can be done through OR-AND as well as NOR-NOR. And same hardware can provide negative logic based realization with an alternative interpretation of high and low voltages.

Thank you.