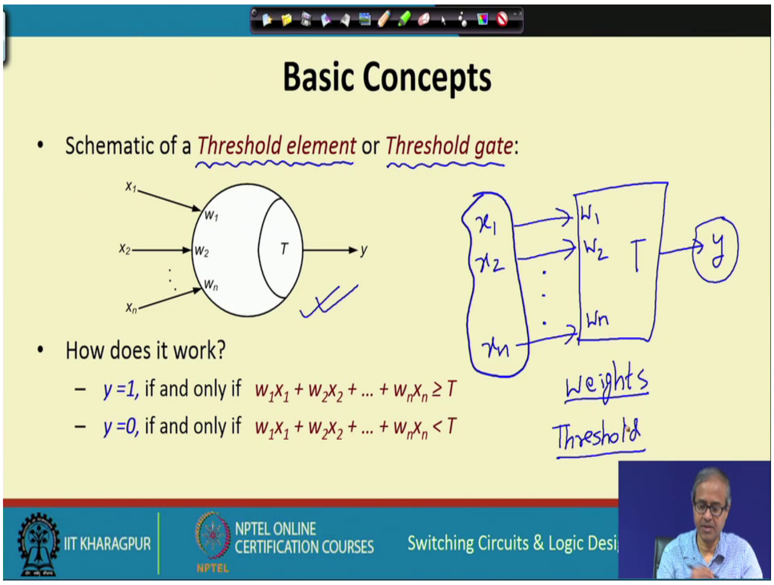
Switching Circuits and Logic Design
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 30
Threshold Logic and Threshold Gates

So, far if you recall you had discussed various ways designing logic circuits using conventional gates like and OR, NOT, NAND, NOT, XOR and also some functional blocks like multiplexer decoders etcetera. In this lecture will shall be looking at a slightly different way of designing logic functions, not using the conventional gates, but using something which is a little unconventional. The title of a talk here is this lecture is Threshold Logic and Threshold Gates. So, in this lecture we shall be talking about something called threshold logic circuits and how we can implement threshold logic circuits using threshold gates.

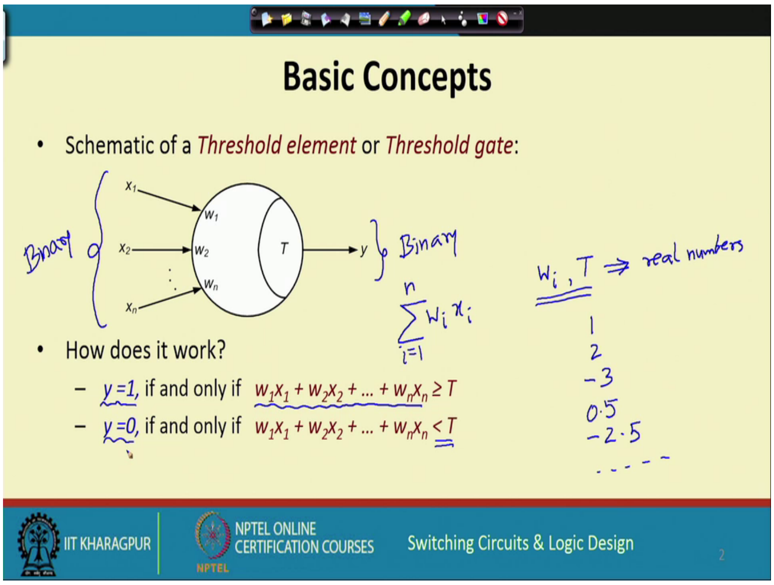
(Refer Slide Time: 01:14)



So, let us see the basic idea behind this so, called threshold logic ok. The first thing is that, in threshold logic the basic element that you talk about is called a threshold element or a threshold gate this is how a threshold gate looks like in its schematic form. So, you see in a threshold gate what are the basic components, if I show it as a rectangular box there are some inputs, which are applied. So, in this example have I shown that there are n inputs x 1, x 2 up to x n and there is one output y.

Now, there are a few other parameters you can see each input assigned some weight is W 1, W 2, W n these are called weights and, there is another parameter quality is called T, T is called a threshold so, unlike a gate which we have learned earlier and or not etcetera, where only inputs and the output matters, here in addition you have some weights and threshold. Now, how does the weights and threshold play a role? You say this threshold element, or threshold gate the weight works well the inputs x 1, x n these are binary inputs, means they can be 0 or 1.

(Refer Slide Time: 02:55)

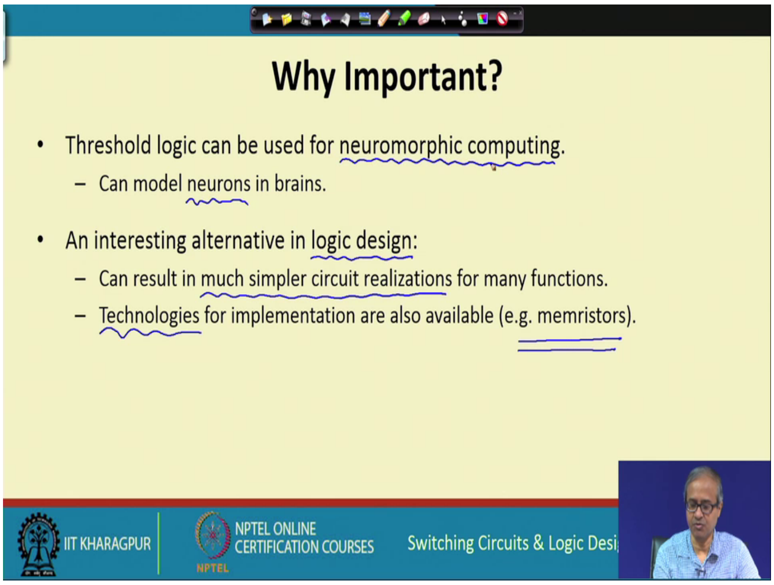


The output y this is also a binary output, but this weights W i and the threshold these are not binary numbers are real numbers. Real number means they can be integers, they can be fractions, they can be negative also, they can be any arbitrary numbers like for example, is weights and threshold values can be 1, they can be 2, they can be minus 3, 0.5 minus 2.5 etcetera. So, you can have any such numbers integer numbers not only integers also numbers with fractional points both positive and negative.

Now, the way the output get us decided is as follows the output why you will be equal to 1, if the weighted sum of the inputs by weighted some inputs by weighted sum means summation W i, x i i equal to 1 to n.

So, you see here we have written down in the expanded form W 1, X 1, W 2, X 2 upto W n X n if this weighted sum is greater than or equal to the threshold T, then the output will be 1, otherwise the output will be 0. This is how a basic threshold gate works. So, you have the inputs you have the weights assigned to the input you have a threshold, you compute the weighted sum, if the weighted sum is greater than or equal to the threshold the output will be 1, if it is less than the threshold the output will be 0, this is the functional behavior of a threshold gate ok.

(Refer Slide Time: 04:56)

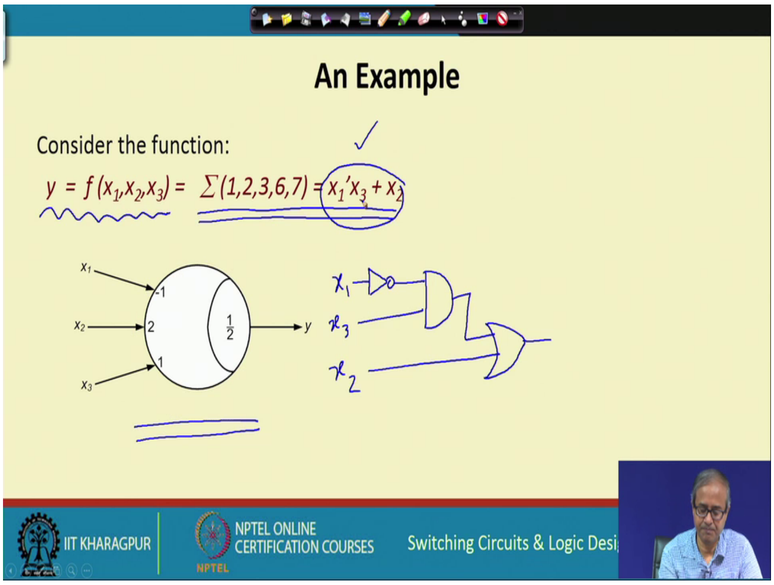


Now, let us try to understand why threshold logical and threshold gates are considered important today. The first thing is that threshold logic has a direct connection with neuromorphic computing, well neuromorphic computing is a branch of you can say computer science, where we are trying to mimic the behavior of a brain. The way the brain works there are basic building blocks in the brain which are called neurons, neurons work in a way which is very similar to threshold gate, there are weights there is a concept of a weighted sum, there is a concept of a threshold, this is how people normally model a neuron.

So, if you have a way to build a threshold gate, then using threshold gifts you can possibly model the brain also so, this is called neuromorphic computing. So, you can model the neurons that are present in the brains. Well if you forget neuromorphic computing even in conventional logic design, this can be considered as an alternative. Because we shall see a few examples that you can get much simpler circuit realizations, for many functions not all functions of course, for many functions using threshold gates you can get very small circuits, very easy to design an implement ok, this can be one advantage.

And second thing is that although threshold logic is relatively old, if this is not a new concept. But technologies for implementing them efficiently is available only recently, earlier the technology was not available that is why people did not build circuits using threshold gates, but today people are thinking about building. For example, we talked about memory stress earlier, using memory stress people are working very actively toward building are threshold logic gates and with applications in neuromorphic computing fine.

(Refer Slide Time: 07:20)



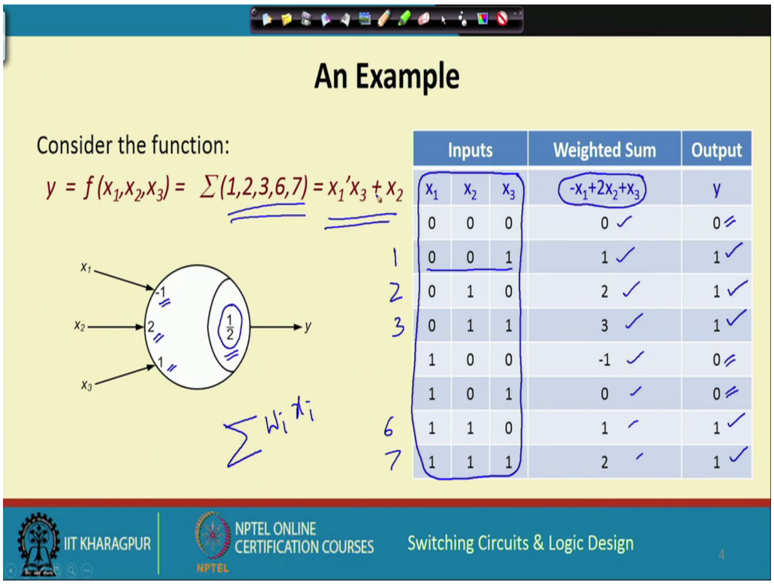
Let us take an example now here we considered a four variable function, let us look from the other side. Let us say that we want to implement a function like this. So, it is a sum or product specification by the true min terms are 1 2 3 6 and 7. So, if you do a minimization using k map for any other method, you will see that will minimum form is this x 1 bar x 3 or x 2.

Now, here I am not telling how, but this is a threshold logic implementation of this function. Now, you look at one thing if you wanted to implement this function using conventional gates, then you would require an AND gate with x 1 you would require a not gate, x 1 bar x 3 and this will go to an OR gate, it will be connecting x 2 here.

So, you would be requiring three gates, but using threshold logic this is just an example single threshold element you can implement this function.

Now, let us see how this single threshold element actual implement this function you see, here the weights that are used are minus 1 2 and plus 1 and the threshold is half. Let us see how this gate works.

(Refer Slide Time: 09:07)

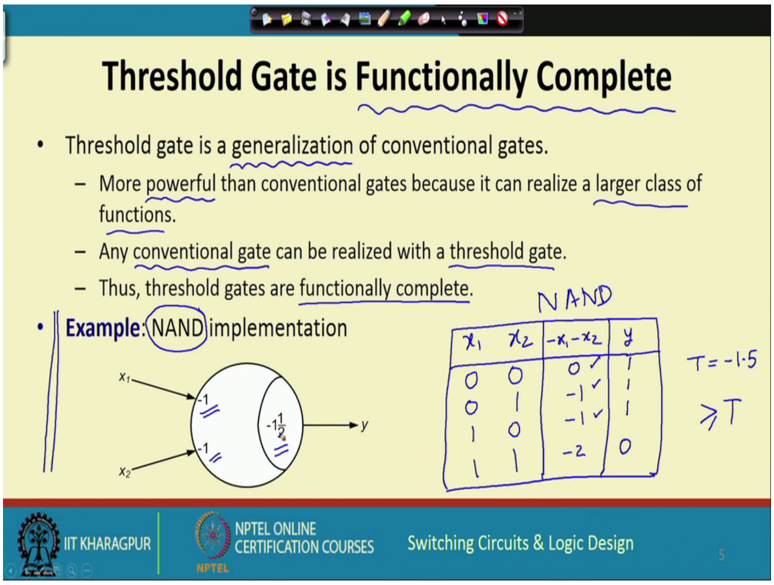


Look at this truth table well on one side; we have considered all possible input combination of x 1 x 2 x 3. And because the weights are minus 1 2 and 1, we are computing the weighted sum as I said weight multiplied by the inputs. So, here the expression will be minus x 1 plus 2 x 2 plus x 3, which is this so, with respect to the values of x 1, x 2, x 3.

If you if you calculate this expression the value that you get are like this 0 1 2 3 minus 1 0 1 2, you can check for example, for 0 1 1 it will be 0 plus 2 into 1 plus 1 which is 3 minus 1 1 0 0 minus 1 plus 0 plus 0 minus 1.

So, now you see because the threshold is 0.5 half. So, among these how many of them are greater than equal to half and how many of them and less than half, you see this is greater than half, this is greater than half 2, 3 is also greater than half 1 is also greater than half 2 is also greater than half, but this 1 0 minus 1 and 0, these are not greater than half that is why you are getting the output 0. So, you can verify this is nothing, but what you wanted to implement 1 2 3 6 7 you see this is this one corresponds to 0 0 1 which is 1, this is 1 2 0 1 0 this is 2 1 0 1 1, this is 3 1 1 1 is 0 is 6 and 1 1 1 is 7, you see 1 2 3 6 7 this is what you wanted to implement ok. So, this example shows that how using a single threshold element OR gate we can implement a complex switching function a switching expression.

(Refer Slide Time: 11:22)



The next thing that you want to talk about is well we talked about functional completeness earlier, we said that when you build circuits, we should be confident that the basic building blocks that are given to us those can be used to design an implement any circuit to you want for instance AND OR NOT by definition they are functionally complete. So, if you have these three kind of gates given to us we can design any circuit, similarly we proved NAND is functionally complete NOR is functionally complete. So, if you give us only NAND gates we can design any circuit using only NAND gates.

Now, here we try to show that is threshold gate also a functionally complete gate. So, using threshold gate alone therefore, we can design any function you want let us see what is our logic. So, we are trying to say threshold gate is functionally complete. The first thing is that this already have seen through the example that we showed just, now threshold gate is some kind of a generalization of conventional gates, they are more powerful. Because we can implement functions which otherwise would be requiring several conventional gate, we can implement the same function using a single threshold element, or a single threshold gate in that way a threshold gate is more powerful, because it can realize a larger class of functions.

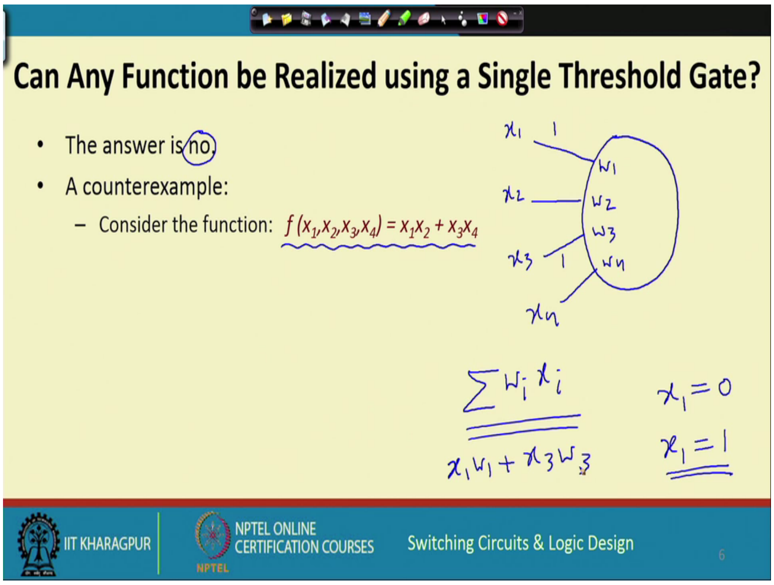
Now, the point you want to show here is that because we are saying functionally complete, we want to show that any conventional gate should be realized with a threshold gate. So, if you can show this then you can prove that threshold gates are functionally complete. Now, the way we give the proof is through an example, we show a threshold gate implementation with the input weights minus 1 and minus 1 and the threshold of minus 1.5 minus 1 and half.

So, you can verify that this threshold gate actually implements the NAND is function and, because NAND is functionally complete. So, if you can show that this threshold gates actually implements NAND so, we have proved that threshold gate is functionally complete.

Now, let us see how this implements NAND, let us quickly again draw the truth table here so, x 1 and x 2 are the inputs so, there can be four combinations 0 0 0 1 1 0 and 1 1. And here what is the weighted sum minus 1 minus 1 so, simply minus x 1 minus x 2 so, the weighted sum is minus x 1 minus x 2. So, if you just calculate the weighted sum on these it will be 0 minus 1 minus 1 and minus 2.

So, what will be the output y the threshold here, the threshold is minus 1.5. So, if this weighted sum is greater than or equal to T, then the output will be 1. So, in this case you see minus 1.5 the first three they are greater than minus 1.5 minus 2 is less than minus 1.5. So, for the first three the output will be one this the output will be 0 this is nothing, but the NAND function right. So, we have proved that using threshold gate we can construct a NAND gate so, therefore, it is functionally complete ok.

(Refer Slide Time: 15:38)



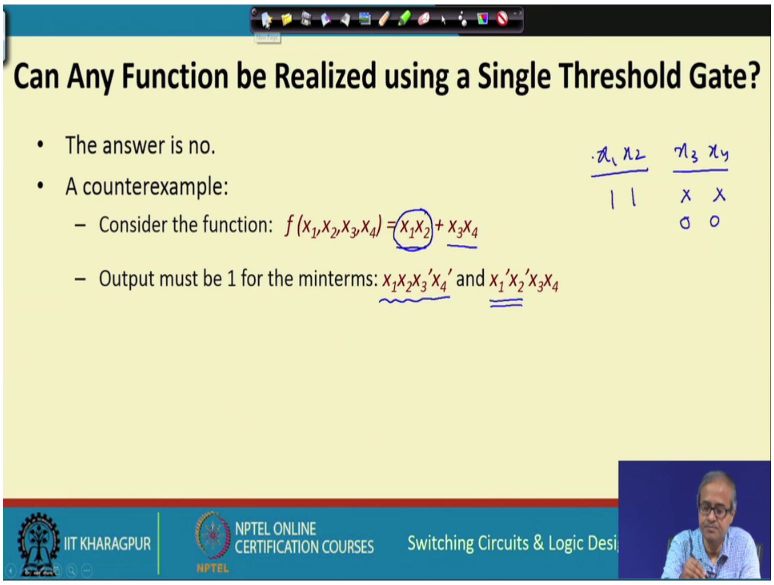
Now, the next question we ask is, can we implement any arbitrary function using a single threshold gate, well earlier we have seen that that implemented a slightly complex function, say x 1 bar x 3 plus x 2 that kind of function using a single gate, now the question is given any arbitrary function can we implement using a single thresh old gate.

Let us try to answer this question of course; the answer is no all functions cannot be implemented. Now, how do we prove this? We give a counter example, we show an example logic function and give a justification why is this cannot be implemented using threshold logic.

Let us see so, the first thing is that the answer is no as I said so, you cannot implement any function arbitrary function using threshold hate. Now, the example with we take is like this now one thing we would like to just tell you, you just understand how a threshold gate works once more the inputs are x 1 x 2 x 3 and x 4, with the weights W 1 W 2 W 3 and W 4.

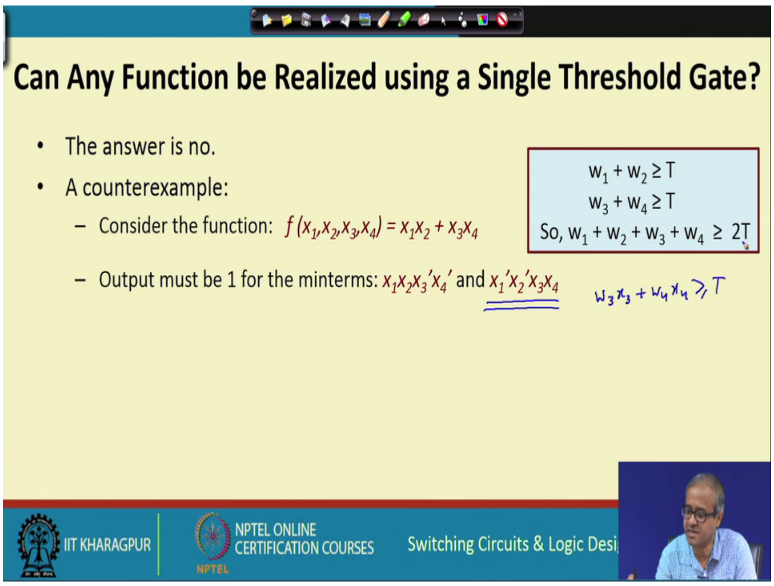
Now, when you calculate the weighted sum right if, and some particular input let us say x 1 is 0, then that component x 1 W 1 will not appear in the weighted sum this will become 0. So, you will have to consider only those combinations for which it is 1, only those terms will come for example, if x is 1 then x 3 is 1, then the weighted sum will be only x 1 W 1 plus x 3 W 3, x 2 and x 4 will not appear here, because they are 0 right ok.

(Refer Slide Time: 17:55)



So, with this example because the product terms here are x 1 x 2 and x 3 x 4. So, the output must be one for these two minterms so, x 1 x 2 say x 1 x 2 must be 1 must be 1, but x 3 x 4 can be anything they can be 1 x 3 x 4 are actually do not care, then the first product term will be 1, but because we are talking about threshold gate, we are talking about the minimum what is the minimum weighted sum minimum weighted sum will happen, if this 2 as 0 0. So, let us take the 0 0 part x 3 bar x 4 bar. Similarly 4 x 4 we take x 1 bar x 2 bar, let us take these two minterms.

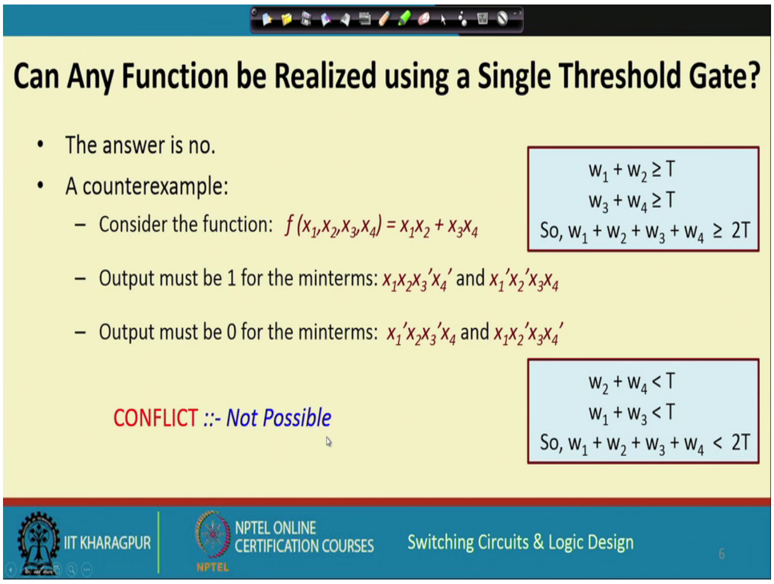
(Refer Slide Time: 18:49)



So, if we take these two min terms the output is supposed to be 1 so, what does this mean just in terms of weighted sum it will be x 1 W 1 plus x 2 W 2 because these two are 0, they will not come and because x 1 x 2 are 1 and 1 it will only W 1 plus W 2. Now, in order the output be one this must be greater than equal to the threshold.

Similarly for the second min term x 3 x 4 are 1 1 so, W 3 plus W 4 must be greater than or equal to T, because your condition was W 3 x 3 plus W 4 x 4 greater than or equal to T, now x 3 and x 4 are 1 so, it is W 3 plus W 4 greater than or equal to T. So, if you add this to up you get W 1 W 2 W 3 W 4 their is greater than or equal to twice T right.

(Refer Slide Time: 19:55)

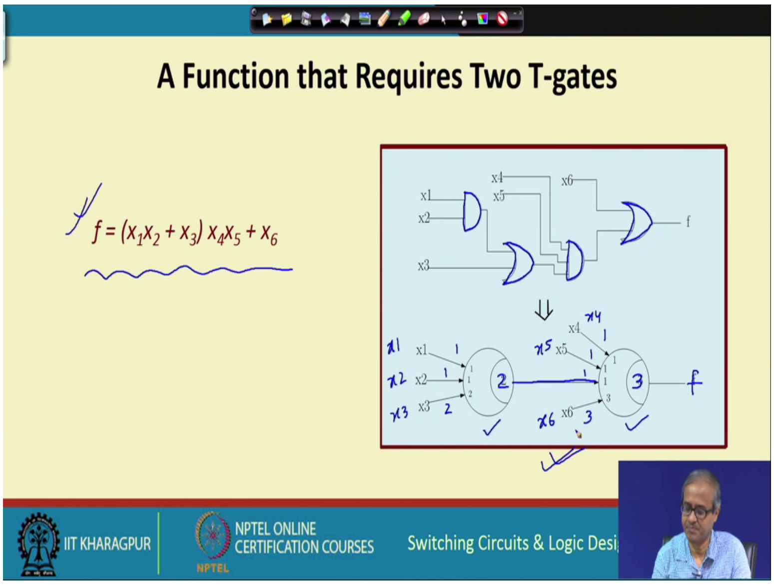


Now, let us look at the faults min terms, the output will be 0 for which min term now the reverse you see output will be 1, if either x 1 x 2 is 1 or x 3 x 4 is 1. Now, output will be 0, if one of x 1 x 2 and one of x 3 x 4 as 0.

So, let us look at two conditions here x 1 is 0 and x 3 is 0 here x 2 is 0 and x 4 is 0, just let us take these two so, others also you can take and you can make a similar justification. So, if you take this then you can arrive at conditions like this, because for this case your output is supposed to be 0 so, the weighted sum must be less than the threshold for this case, because x 2 and f 4 are 1 so, W 2 plus W 4 must be less than T and for the second case x 1 and x 3 W 1 plus W 3 must be less than T.

So, if you add them up you see their sum is less than twice T. So, you see you have arrived at a contradiction so, in one side your saying that the sum of the words must be greater than equal to 2 T on the other side you are saying it must be less than 2 T. So, the overall conclusion is we cannot find the value of weights W 1 W 2 W 3 W 4 such that these conditions are satisfied simultaneously. So, we can say that this function cannot be implemented using a single threshold gate. So, we have actually showed this as a counter example, because we have a counterexample. So, this condition is not true fine.

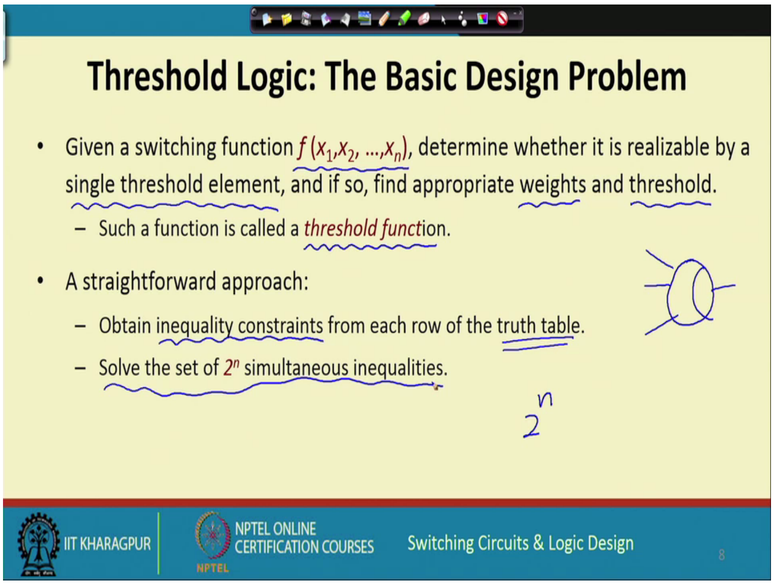
(Refer Slide Time: 21:49)



Let us take another example where is showing a function, that requires more than one threshold gates for implementation. So, here I am just showing the final solution not going to this steps, this is a function of variables so, the lines faint I am actually showing the gates in using convectional logic, we will be requiring four gates to implement 2 AND gates and 2 OR gates.

Now, what we are showing here is that using threshold logic gates, you can have two threshold logic gates to implement the same functions. So, you can actually verify that it works. So, I live in a exercise for you for the first gates the inputs you are applying are x 1 x 2 and x 3 this inputs are x 1 x 2 and x 3 the weights are 1, 1 and 2 and the threshold is 2. For the second gate the inputs are x 4 x 5 the output of this gate is also an input and x 6 and the weights are 1 1 1 and 3 and the threshold is 3. So, the function f that you get will correspond to this function, this you can verify. So, this is one function which can be implemented using two threshold gate and this is an example realization of that right this you can check.

(Refer Slide Time: 23:34)

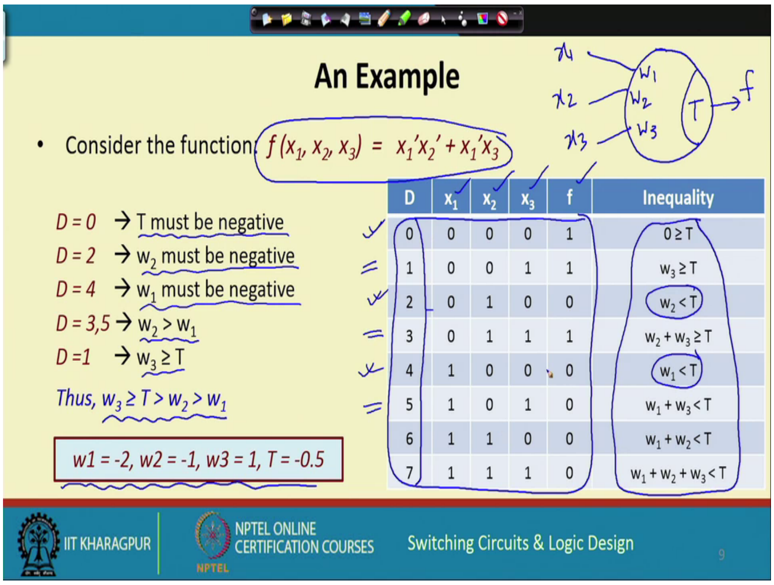


So, the basic design problem in threshold gate is given any arbitrary switching function let us of n variables, the main question you want to answer is to determine, whether it can be realized by a single threshold gate by a single threshold element and, if it is possible what should be the weights what should be the threshold value. So, the design problem of a threshold element or a threshold gate is to determine, or find out what will be the value of my weights what will be the value of the threshold. So, once we have to do this your synthesis problem is solved ok.

So, if a function can be implemented using a single threshold element, we say that it is a threshold function. So, a threshold function is a function which can be implemented by a single threshold gate like this.

So, we shall take an example to show how we can systematically check for this and how to determine the weights. So, we construct the truth table and, from every row the truth table we obtain some inequality constraints we shall see how. So, we will be having 2 to the power n such constraints, because there are two to the power n rows in the truth table. So, will have to solve those 2 to the power n inequalities at the end right, this is the basic idea.

(Refer Slide Time: 25:15)



Let us take an example to work this out, we consider this function and for this function, we have constructed the truth table. So, we are showing the inputs x 1 x 2 x 3 the output f and also the decimal equivalent of the inputs 0 1 2 3 4 5 6 7 8. Now, suppose I have I mean I am that I can build a threshold gate to implement this function. So, the inputs will be x 1 x 2 and x 3, my weight will be W 1 W 2 and W 3 there will be a threshold three and the output will be f.

So, if I have input 0 0 0 so, my weighted sum will be 0, because the output is 1 that weighted sum must be greater than equal to T. Similarly for 0 0 1 the weighted sum will be x 3 W 3 only W 3 this is 1 so, W 3 also must be greater than equal to T for the row number 2 0 1 0 only x 2 is 1 so, W 2 it is 0 so, W 2 must be less than T that is why it is 0 here, both x 2 and x 3 are 1 so, W 2 plus W 3 this is one should be greater than or equal to T, here W 1 0 less than equal to T W 1 plus W 3 0 less than or equal to T W 1 plus W 2 0 again less than T and all 1 it is 0. So, their sum also should be less than T.

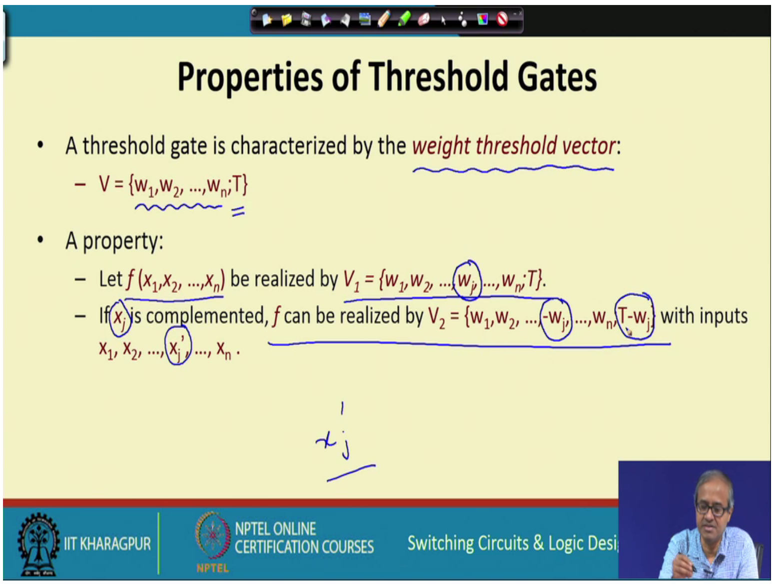
So, you see we have got 8 inequality constraints directly from the truth table from here, we have to solve for W 1 W 2 W 3 well, there are ways of solutions and, showing you just intuitively how you can go about doing it D equal to 0 if you row it says T should be negative. So, the first conclusion is T must be negative look at row 2 D equal 2 it says W T is less than T, because T is negative. So, W t also must be negative and should be less than T. Similarly row number four says W 1 less than T, because T is negative W 1 should also be negative.

So, there is you can see directly and fro rows 3 and 5, what you can see it says W 2 plus and W 3 is greater than equal to T and W 1 plus W 3 is greater than equal to T so, W 3 cancels out less than T it says W t greater than, or equal to T W 1 less than T so, from that you can reduce W 2 greater than W 1 well understanding that W 1 and W t are all negative and lastly from the D equal to 1 row W 3 is greater than equal T this is another condition.

So, combining this you have, these conditions to be satisfied that W 3 should be greater than equal to T, T must be you have this conditions T must be greater than W 2 greater than W 1, because this condition are there T must be greater than W 2 T must be greater than W 1 and already we have shown W 2 must be greater than the 1.

So, you can have any arbitrary choice that satisfies this and of course, T must be negative so, one possible choice I am showing here, there are infinite possible solutions so, one solution is W 1 is minus 2 W 2 is minus 1 W 3 is plus 1 and T is minus 0.5, you see all this conditions satisfied W 3 greater than equal T, T greater than or equal to W 2, W 2 greater than equal to W 1 ok. So, this is how given a function given a truth table well, if it is a threshold function you can reduce the values of the weights and the threshold ok.

(Refer Slide Time: 29:38)

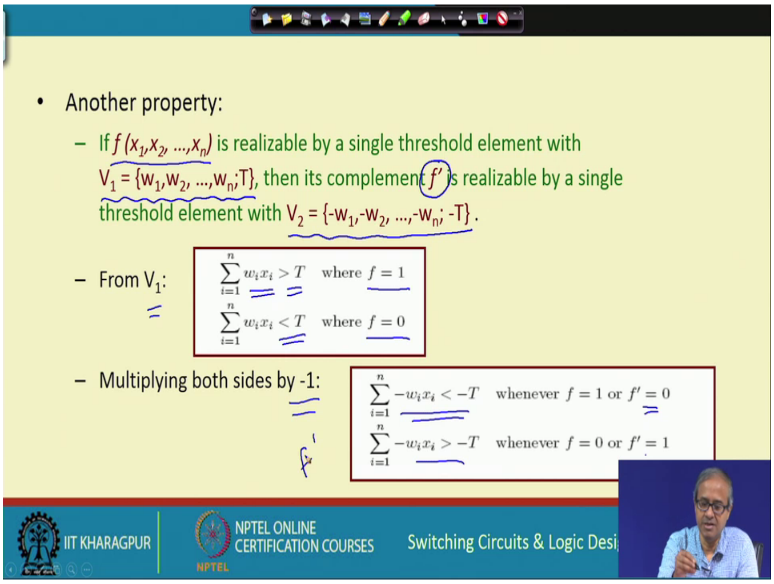


Now, we shall look at a couple of properties of threshold gates, or threshold functions. The first property says that any threshold gate as you know, it is characterized by the weights of the inputs and also the threshold, they taken together is called weight threshold vector. So, any threshold gate is characterized by its weight threshold vector.

So, consider a function f, which can be realized by weight threshold vector W 1 W 2 W n and T. Let us call it if you want let us say if we have any particular variable x j have any particular variable exchange the corresponding what is W j the corresponding weight is W j.

So, if we compliment x j so, instead of x j in the function we make x j bar, then the corresponding function can be implemented only by negating the weight of that W j and changing the threshold from T to T minus W j, this can be proved ok. So, given a function if you want to realize another function for one of the inputs are complimented, then you can just adjust the weights like this is one result I wanted to show.

(Refer Slide Time: 30:55)



And another result is let us considered this f is a threshold function and this is the corresponding weight threshold vector.

Now, suppose we want to implement the complement of f, it says the compliment function can be directly realized using the weight threshold vector, where you negate all the weights and the thresholds minus W 1 minus W 2 minus W n and minus T, well it is very easy to prove the this, because we see from the first condition from V 1, because this is a threshold gate the weighted sum of W n x i must be greater than or equal to T, whenever f is 1.

And it should be less than T, whenever f is 0. Now, if you multiply both sides of this inequalities by minus 1 so, this W i x i will become minus W i x i t will become minus T and, this greater than will become less than less than will become greater than right. So, from here you can see that if you make the weights and threshold negative and, this less than greater than gets reversed. So, so in that earlier case, if you was 1 now it is becoming 0.

Now, earlier case it was 0, now it is becoming 1 so, it is actually the complement function right. So, we have seen a few such properties there are many such properties of threshold functions so, so we are not going to go into the detail of that any of you are interested, there are some very nice literature available and books available on threshold logic functions you can go through them, but I just wanted to give you a brief overview about threshold logic and threshold gates and how they can be used to implement switching functions.

Thank you.