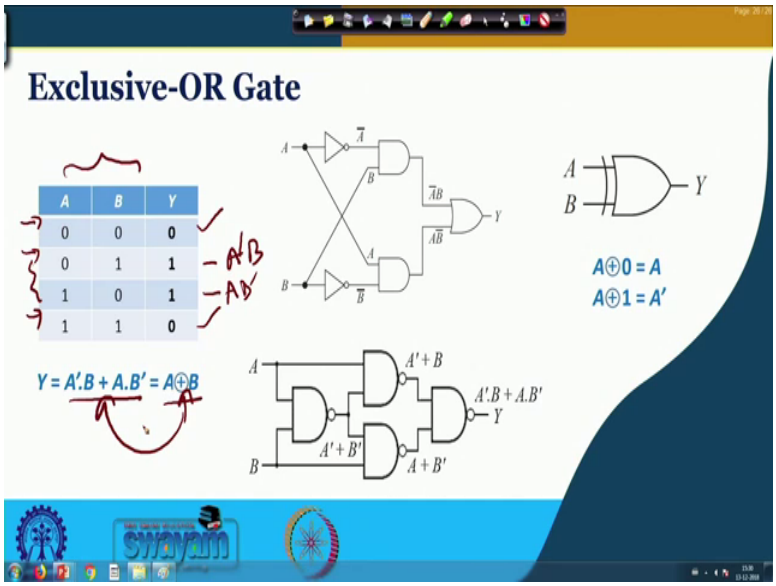
Digital Electronic Circuits
Prof. Goutam Saha
Department of E & E C Engineering
Indian Institute of Technology, Kharagpur

Lecture – 20
Parity Generator and Checker

(Refer Slide Time: 00:26)

Hello everybody. In today's class, we shall discuss Parity Generation and Parity Checking. What is parity, and how is it useful that we shall also discuss in this particular class. So, we shall begin with a discussion on exclusive-OR gate, and then we shall go on discussing the parity, its generation and checking.

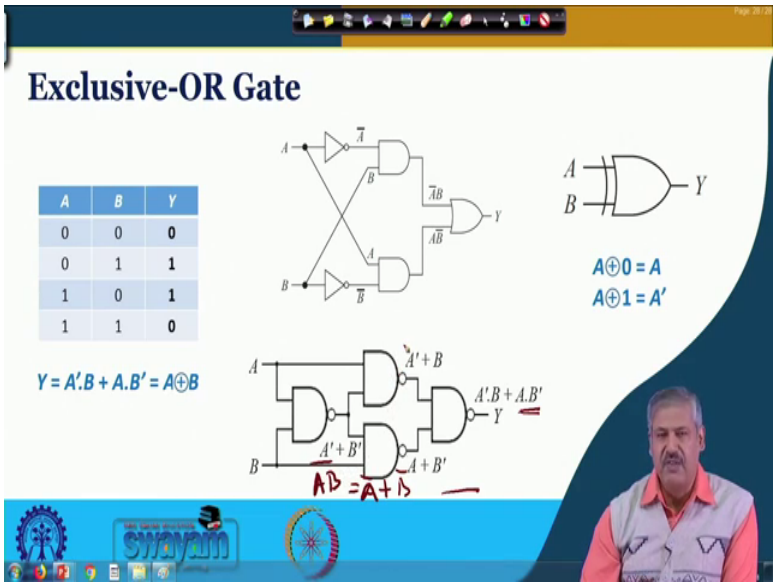
(Refer Slide Time: 00:36)



Exclusive-OR gate follows a truth table as you see over here. When this is a two input a exclusive-OR gate, when A and B both are of same value say 00 and 11, then the output is 0 ok. And when one is 0, and the other one is 1, these two cases the output is 1 this is what is exclusive-OR truth table. And how we represent it, we represent this as A prime B product term, and this is as AB prime. So, it simply sum them up, and then we will get the exclusive-OR function logic function.

And the symbol that we used for exclusive-OR is A, Ex-OR B, I mean this is one particular symbol which is equivalent to what you see over here for simplicity, for a compact representation, we can use this symbol ok. Realization is very simple, so you will be generating A prime B, and AB prime through NOT gate and AND gate, and then we will sum it up. And if you are interested to realize it using only one type of universal logic gate, which is NAND gate or NOR gate.

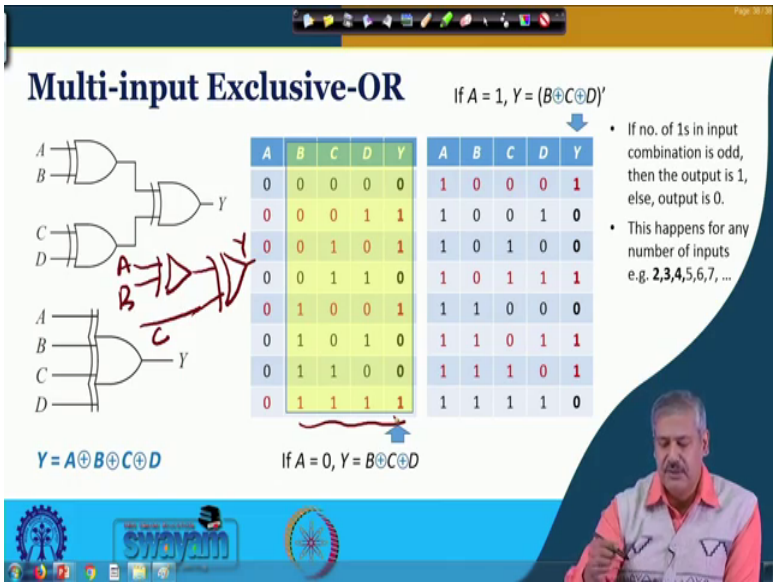
(Refer Slide Time: 02:03)



So, NAND gate is useful, we can use four NAND gate a combination like this by which we can get A prime B, A prime B plus AB prime over here ok. You can check it yourself by so basically this is A prime B prime is AB prime you get AND gate. So, from de Morgan's theorem A prime A prime plus B prime. So, you can go on breaking it up in each of these cases this in NAND function, and then we shall see that this is what you get ok.

So, exclusive or logic gate has got a symbol. So, this is the symbol for two input exclusive-OR logic gate. And in this case, we know from this truth table if any of these input is say B is a 0, see we put it 0 ok, then output follows the input. What is this case, this is the case B is equal to 0 B is equal to 0. So, whenever A is 0, output is 0; A is 1, output is 1 ok. And if B is equal to 1 that is 5 volt if you connect ok, what is this is case, this is these two cases are there. So, when A is 0, B is 1, Y is 1, A is 1, Y is equal to 0, so the opposite of it. So, A Ex-OR 1 is A prime ok, so this we take note of ok.

(Refer Slide Time: 03:29)



Now, multi-input exclusive-OR gate; this is usefully a in our parity generator and checker circuit, which we shall see later. So, multi-input exclusive-OR gate, we look at one example where A, B, C, D 4 inputs are there. So, AB Ex-OR over here CD Ex-OR over here using the two input two exclusive-OR logic function whatever way you will realize it similarly, and then Y is realized by these two ok. So, this is a four input exclusive-OR gate, where I mean symbol as a symbol we can represent it in this manner ok.

So, what is the corresponding truth table, so this is the four-variable truth table. So, 4 2 to the power 4 16 possible combinations of the input will be there. So, in the beginning we see where A is all 0 ok. So, if A is 0, just before we have seen one of the input is 0, then Y follows the other input, so B Ex-OR C Ex-OR D, this is what it follows ok. And if A is equal to 1, it follows the B Ex-OR C Ex-OR D its prime ok, this is what we are just noted.

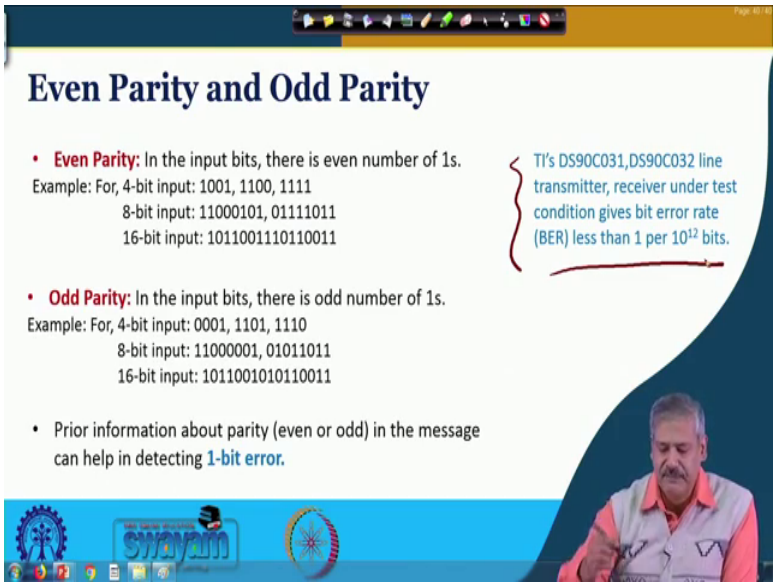
Now, let us look at the truth table. So, if A, B, C, D is 0000, so 00 and 00 over here, so A Ex-OR 00 Ex-OR is 0, this is also 0 the output is 00 Ex-OR is 0 ok. If it is 0001 this length, so 0 0 0 1, this is 0, this is 1, 0 and 1 Ex-OR is output is 1, so you see the output is 1 right. So, this way if you see that if one of the input is 1 out of this 4, the output is 1 ok. So, if two inputs are one, for example let us consider this particular case ok. So, 01 01, so 01 and 01; so, this is 1 and this is 1, so 11 both are of same value. So, the output

will be 0. So, in 11, the output is 0 ok. So, you can take up any other case where there are two 1's see A and D or AB AB and AB are 11. And CD are 00, so 11 and this is 00, so this is 0, this is 0, the output is 0. So, you can look at all possible cases of two 1's appearing, you will find the output is 0.

Now, let us look at three 1's a case like this. So, A is 0, and BCD are all 1. So, this is 1, this is 1, and this is 1. So, 01 this is 1; 11, so this is 0; so, 10 this is 1, so that way if you continue, you will see that whenever the number of input in the combination is odd number of 1's in the input combination is odd, then the output is odd else the output is 0 ok, this you can verify you can check from the truth table.

And this happens for 2, 3, 4, 5, 6 as many number of you know you can have inputs, it will be like this. You can check for yourself a case like this, so three input case, so this is A, this is B, and then this is taken to this is your Y, this is your C, you will find the similar thing happening. So, A is equal to 0, A, B, C, so basically three input case, this is the example, you will see that the exactly this similar thing is happening ok. So, this is we take note of which is which will be usefully in our parity generation and parity checking circuit that we are going to discuss ok.

(Refer Slide Time: 07:40)



Now, we first define what we understand by parity ok. Even parity and odd parity, there are two types of parity. So, even parity means, in the input bits there is even number of 1's that is present ok. So, if you look at the you know number of 1's that are present if

you count them up ok, in that particular group and if you see that it is even number, then you say that that particular group has a even parity has an even parity.

For example, this is example for 4-bit for group you know group of 4 bits, so 1 1, two 1's are there, so this is even parity. Here four 1's are there, it is even parity ok. So, all 0 that is also considered even. Please remember if there is no 1, all 0 that is also considered as even. For 8-bit combination you can see, this is 1, 2, 3, 4, this is even parity. So, this is 4, and 2, 6 this is even parity. Similarly, for 16-bit combination you can count them up, and you can see if it is even, then it is even parity.

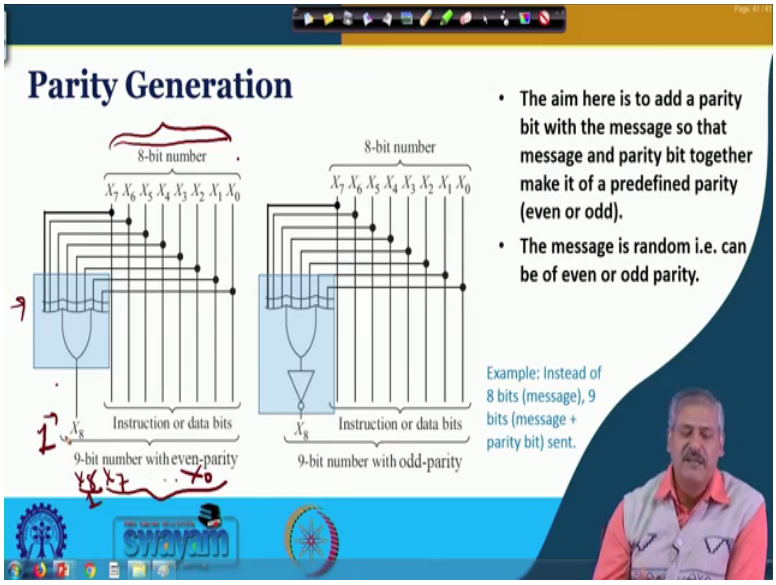
And odd parity is just its you know similar where the input bits if you look at the you know number of 1's that is present if you count them up, and if it generates the odd number, then it is odd parity. So, the examples are for 4-bit cases. This is odd parity, because only one input is present. So, three 1 1 1 is present three 1 is present here, so this is odd parity.

Similarly, the other two cases if you can count them up, for example this is 16-bit input 1, 2, 3, 4, 5, 6, 7, 8, 9 ok, so nine 1's are there, so this is having a odd parity ok. Now, how does it help what is it is I mean if you have a in a particular group even parity or odd parity particular parity in the particular you know, then how can it be useful. So, if we know a priori particular group, which is particular you know bits combination of bit, which is sent from one place two another has got pre-define parity say even parity ok. Then during transmission because of noise, interference or some other thing; if one of the bit becomes erroneous that means, it changes ok. So, one of the 1 becomes a 0, what will happen. The parity will be changing, so what was even parity will now become odd parity ok.

Similarly, if one of the 0 becomes 1 that will also that is also going to happen right. So, if 1-bit changes, then becomes erroneous, we can detect that there is an error. So, what was supposed to be even parity is not even anymore, it has become odd. And similarly for if it was supposed to be odd parity, we can see that it has become even. Now, if more than one changes two changes 2-bit changes, then it will you know come back to the same thing right. So, but the likely hood the probability of more than one bit changing in a group is much much less. When the practical transmission of you know binary digits from one place to another, if we consider that is that probability is very less.

So, at I mean usually you can see a group of you know like this only 1-bit may become erroneous occasionally. But, if you can detect by this method, then you can ask for retransmission or corrected by some other mechanism ok. So, here is an example where (Refer Time: 11:24) instruments transmission line transmitter and receiver design, and it is tested for bit error rate less than 1 per 10 to the power 12 bits, you can imagine that this is this does not happen very often. In certain cases certain kind of noises it might happen, but for that we have got different way to handle that thing which we may study in digital communication course in some other contexts.

(Refer Slide Time: 11:55)



So, we are looking for having a predefined parity and checking it, so that any error because of you know transmission, noise or so we can detect it, and possibly correct it through retransmission or some other means, which we can discuss later ok.

But, the message that we have in the first place is random in nature ok. So, if there is a 8-bit message that means, that message can have even parity, can have odd parity, you can have many many different things ok. So, we do not know for sure that whether it is a having a even parity or odd parity. So, we have to enforce a in that group by some mechanism a specific kind of even parity ok, so that will help us and from the receivers side, it will be known whether a specific you know particular parity is being followed or not ok. So, how we can do it?

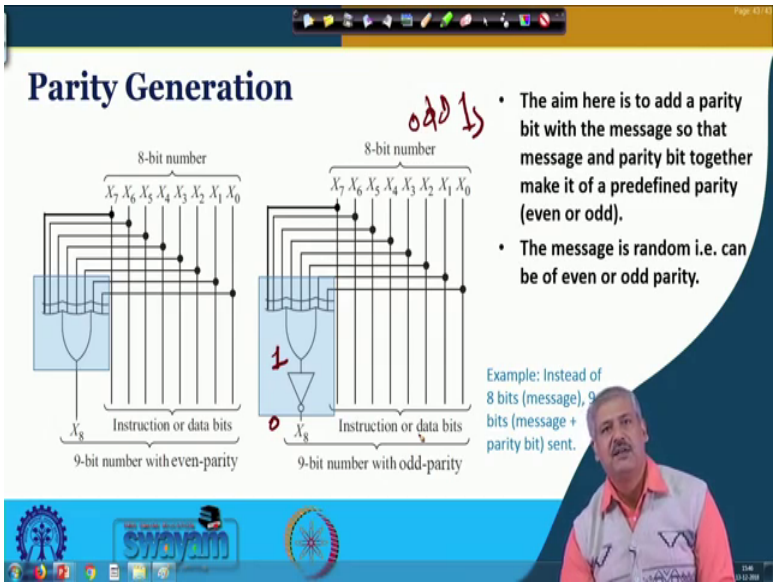
So, one example we can see one way of doing it is these are 8-bit numbers that is message bits X 0 to X 7, so we send it through a 8 input Ex-OR gate 8 input Ex-OR gate. So, what will be the eight outputs output if there is a even parity here, the output will be a 0. And if there is a odd parity that means, number of 1's present is odd, then the output will be 1 that you have checked you have seen Ex-OR gate truth table ok. So, if this is even right, then this is 0.

So, if we considered include X 8 in that particular group, so right now instead of 8-bit, we are sending 9-bits ok, so X 0 to X 7 and additional X 8 as parity bit ok. So, this X 0 to X 7 has even, and X 8 is 0. So, all together it is even fine that is we are that is what we understand. Now, if X 0 to X 7 is odd means, odd number of 1's are present.

And X 8 at the time since it is odd, this will be generating a 1 over here. So, this one and odd number of 1's that is present between from X 0 to X 7 together, it makes even number of 1's. So, either way whether this 8-bits are having odd number of 1's or even number of 1's, this X 0 to X 7 together with X 8, the whole group of 9-bits will always have even parity.

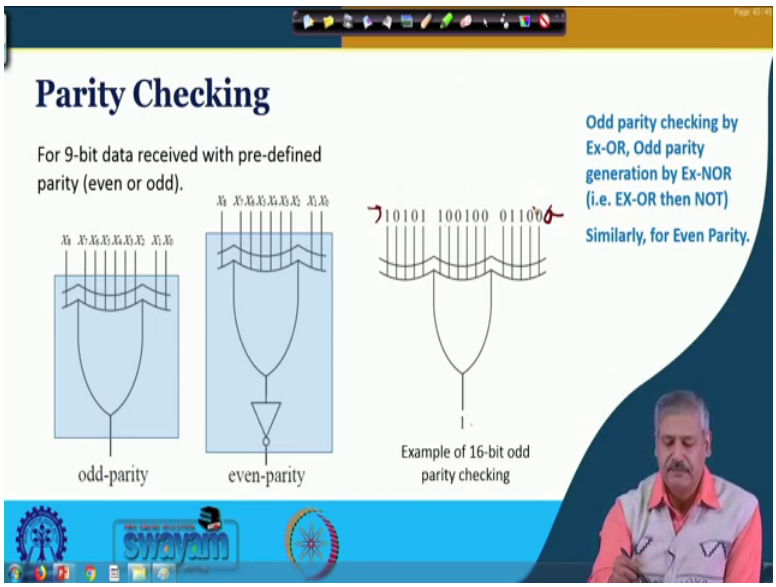
So, this is what we can employ to enforce a even parity in this particular message transmission ok, irrespective of even or odd number of 1's present the group X 0 to X 8 will always have even parity is it fine. So, from the transmitter it is when told, so receiver will know that it is even parity. It will check if it is not even parity ok, then know that this particular group there is an error ok. So, instead of even parity if the transmitter and you know receiver, they decide on odd parity ok. So, then how to generate how to you know ensure that the particular group of you know 9-bits you will be having odd parity. So, what we will you do, we will just put earlier these Ex-OR gate, after that an inverter ok.

(Refer Slide Time: 16:00)



So, what does it what do you know do, so if there is a even number of 1's present even 1's present ok, so this output will be 0 right this output will be 0, and it is inverted, so this output it will be 1. So, even one and another one right. So, ultimately it is a odd number of 1's are there. And if instead if odd number of 1's are present right, so these output will be 1 and or NOT, NOT after inverted it will be 0, so odd 1 and 0, so total it is odd. So, this group will be having a odd parity in from in the group of 9-bits ok. So, this is what you will you know do for parity generation ok.

(Refer Slide Time: 17:03)

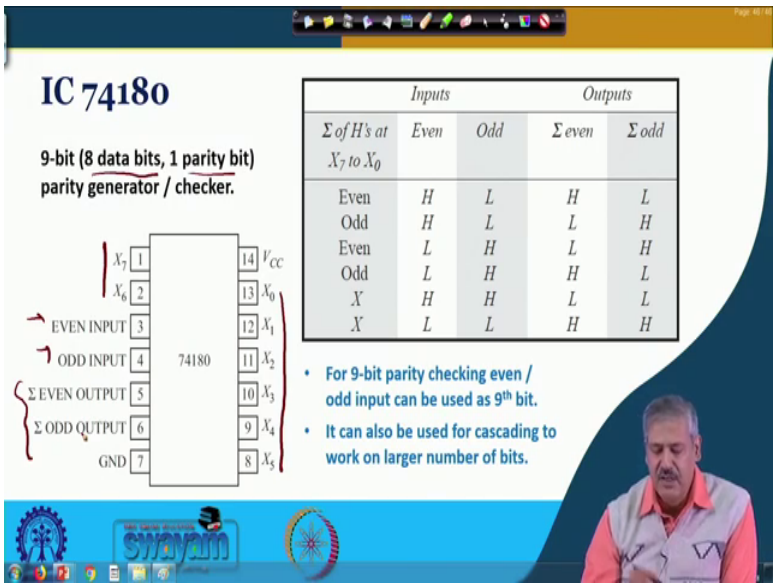


Now, to check parity what will you do. So, similar to what we have done for you know Ex-OR gate. So, for checking odd parity, we shall send it to a 9-bit 9 input Ex-OR gate ok. Say if odd number of 1's are present, this would be this will be generated a 1 this will be generating a 1. And if even number of 1's are present, this would be generating a 0 ok.

And instead of this Ex-OR gate if we put a NOT gate after this, inverter after this, then if odd number of 1's are present this is 1 ok, and after that because of the inversion they will be it is 0. So, if this output is 0, you can say for sure that it is odd number of 1's are present, even number of 1's are not present. Even parity will be detected, when this goes 1. So, this goes 1, when this you know goes 0. And this is 0, when even number of 1's are present over here clear right, it is very simple.

So, same Ex-OR gate which was doing a parity generation can be used for parity checking also just by making a proper understanding of input output you know connection ok. And this is say simple example of you know 16-bit odd parity checking, so these bits are present right. And you can see 1, 2, 3, 4, 5, 6, 7 right, so 6, 7, seven 1's are that the output will be because it is just Ex-OR gate. So, the output, because odd number of 1's are represent 1; so, you can check and you can find that the output is 1. So, odd parity is detected in the input group of 16-bits.

(Refer Slide Time: 18:59)



Now, this is that was the theoretical or conceptual developing part, so IC 74180 is represent as an integrative as a integrated circuit. So, this particular chip actually does

the function of parity generation as well as checking. So, this is a 9-bit out of which 8 or data bit, so 1 is parity bit, the parity generation and checking both it can do ok.

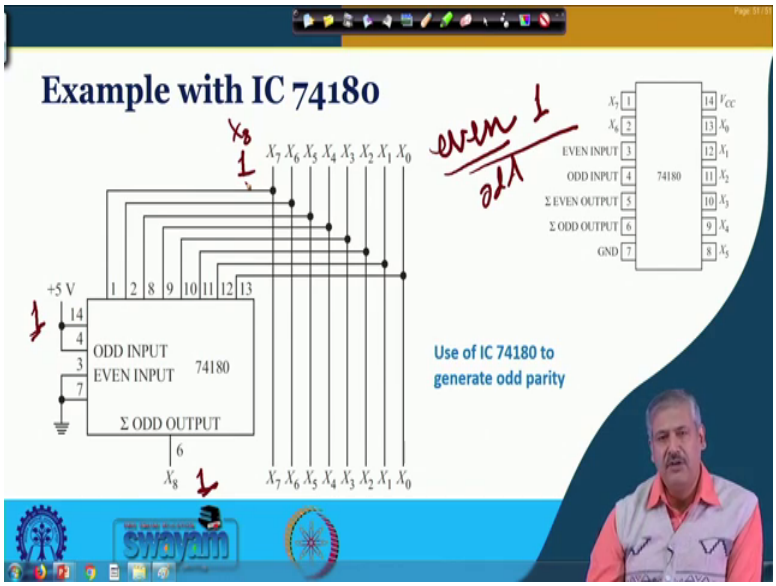
So, you can see that X naught to X 5, and then X 6 to X 7 right, so these are the data bits. And there is a even input and odd input, we shall discuss what it does and this is a summation of even output, summation of odd output ok. So that means, it is basically giving the odd parity, even parity, output define allowance summation means, so either the summation of 1, summation of 0 that is what it is considering. And rest is VCC and ground the power and ground ok. So, it is truth table is like this.

So, when this given input even input, and these odd input. If both of them are high ok, which could not be the case for normal operation is not it. If it is if the input is even from the previous stage, it cannot be odd or if it is odd, it cannot be even, so both of them are high means both it is even, and both it is odd, it is not you know practical, so that is not these two are both of both of them, high it is not being normal operation. So, at the time both the outputs are low. And if both of them are low, the outputs are high. So, this is all truth table is arranged. And irrespective of the other cases what is happening the summations etcetera, you do not bother the output is kept like that ok.

Now, if even input is high that means, the previous stage the input sine it is high and odd is low, so this is a correct proposition, so if one is high, another has to be low right. And if the summation of if summation so basically, then the previous stage the input sign it is said that it is even parity ok. So, if the summation of ones each means ones in 0 to X 7 is even right, so input is even, this is also even ok. So, then the what will be the output, output will be even parity will be detected. It even parity will be high, and odd parity will be low. So, this even parity output will be high, and odd output will be low is understood.

And input is even, and summation of bits is odd, then output will be odd ok. And if input is odd if input is it is odd, and summation of ones is even, then odd and even together the output will be odd ok. And if input is odd, and submission of 1's is odd, then put together the output will be number of 1's, it will be considered as even. So, this is how the truth table is of IC 74180 works out ok.

(Refer Slide Time: 22:40)



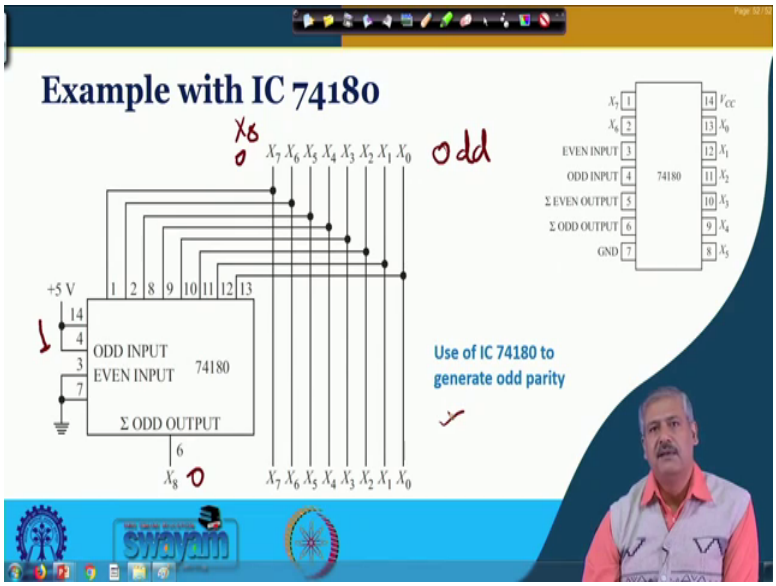
So, let us see how IC 74180 in practice works one example. So, in this case what we see is the X naught to X 7 are the data in which we are checking the parity. So, data is transmitted, we are just the tapping it and putting it to the IC 74180 data input ok. And we have put odd input to the 5 volt, so it is considered this is 1 and even input to ground ok.

So, what you have checking now, by this if we check, we are checking if you look at the odd output, what will you see? So, if this is even summation of 1's are even ok, and since odd is kept here odd input is kept as high, so even and this is odd. So, even number of 1's here, and odd is there from the previous stage odd from the inputs side say even in all together it is odd ok, so this will be high ok.

And if this is if this is odd ok, so odd and odd this is one, so put together it will be 0 right. So, this is what it is doing, when we are talking about it the checking of it ok. And when it does the generation part, checking part we have understood.

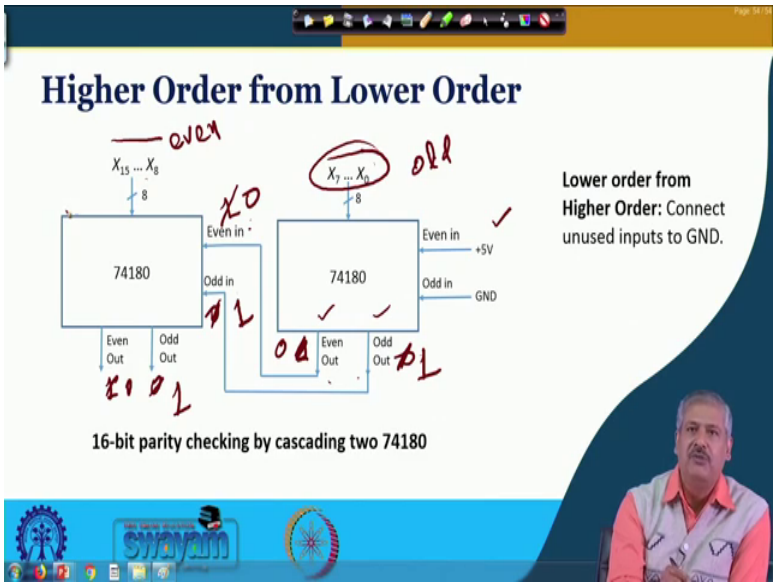
So, when it does the generation part, if this is even right, because this is odd so it will make it odd. So, this even and this one together, it will be odd parity put together all of them together is it clear. Even number of 1's are there and this is one because of which the output will be odd. So, if this X 8 is included here X 8 is included here, so this is even and this is one, so together it will be odd ok.

(Refer Slide Time: 25:08)



And if this is odd, and this odd is high, so this is 0. So, X 8 now is 0, so this is odd and this is you know 0, so put together, it will be odd. So, this can be used to generate odd parity in that sense checking as well as generation both are possible in this manner right.

(Refer Slide Time: 25:40)



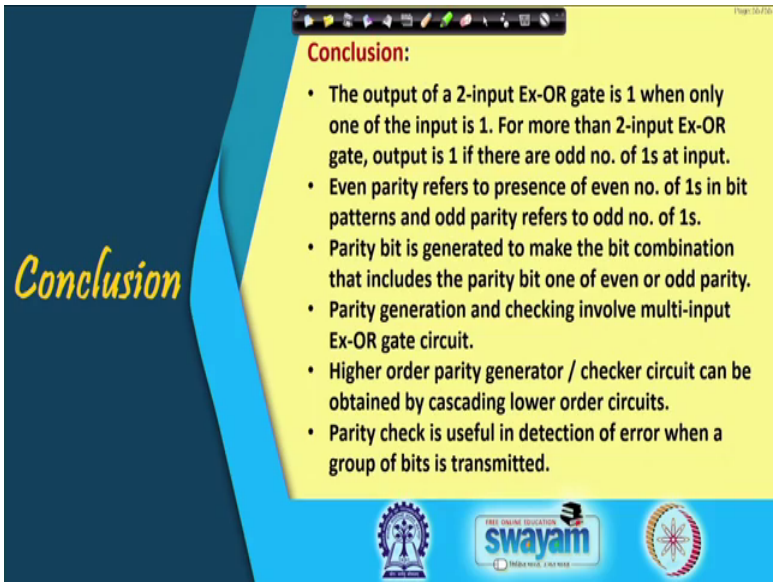
So, we are saying that if we are I mean the IC 71480 can be used in cascade also, this even input or input that can be utilized properly. So, this is also getting harder from you know multiple lower order. So, basically we are in this case, we are looking at checking 16-bit parity ok. So, you are getting a 16-bit parity checking done by using to IC 74180.

So, this is X naught to X 7, and this is X 8 to X 15 alright. So, this is the lower order and this is a higher order, put together it is a 16-bit parity checking that is done in this example ok. So, to start with you are making with even the input side, because there is no for you know previous stage. So, even is considered as high, and odd is considered does not low ok. So, whatever is the number of 1's here that is going to decide the final output. So, in this stage whether it is even or odd that is decided by these two within this particular group whether it is even or odd, it is decided by these two output. See if it is even number of 1's are present, this will be high and this will be low so this is high and this is low at that time.

Now, in this particular group you may 8 to 15, if it is also even, so even and this is even in high, so the output will be this one will high, and this one will low right. But, if in this particular group, this is odd say example this is odd, so this is 0, and this is 1, so this will be 1, and this will be 0, and this is even. So, as a whole with 16-bits will be of odd in odd parity, because odd it is X not to X 7 odd number of 1's are present, X 8 to X 15 even number of 1's present. So, total is odd number of 1's are present ok, so at that time odd is 1 odd is 1, even in is 0, so this is even. So, you will get put together odd out is 1, and even out is 0 ok. So, this is how it works, and you can add more number of stages to get larger number of this thing.

And if you are looking for a 12-bit parity checking instead of 16-bit right, then the higher 4-bits right X 14, X 15, X 13, and X 12 will connect to ground you will connect to ground. So, there no contribute any ones in the system ok. So, if it is 10-bits similarly you know as many number of higher bits will be connected to the ground or say 11 bits, so will connect higher number of 5 bits to the ground and rest will be subjected to the parity checking. So, X 0 to X 7 will go here, and X 8 to X 10 if you are talking about 11-bit parity checking, X 8 to X 10 will go here, and X 11 to X 15 will be connected to the ground ok, and will take finally from here. So, this is how we make use of parity and generation and checking.

(Refer Slide Time: 29:27)



So, to summarize; the output of a 2-input Ex-OR gate is 1 when only one of the input is 1. For more than 2-input Ex-OR gate output is 1 if there are odd number of 1's present at the input. Even parity refers to presence of even number of 1's in bit patterns and odd parity refers to order number of 1's. Parity bit is added to make the big combination that includes the parity bit one of even or odd parity useful for transmission purpose and error checking and things like that.

And parity and generation and checking involve multi-input Ex-OR gate circuit, the similar in Ex-OR gate circuit can do both generation and checking. High order parity generated checker circuit can be obtained by cascading lower order circuit we have seen one example 2 IC 74180 generating 16-bit parity I mean providing 16-bit parity generation and checking. And parity check is useful intension of a error when a group of bits its transmitted ok, so this is about this class.

Thank you.