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

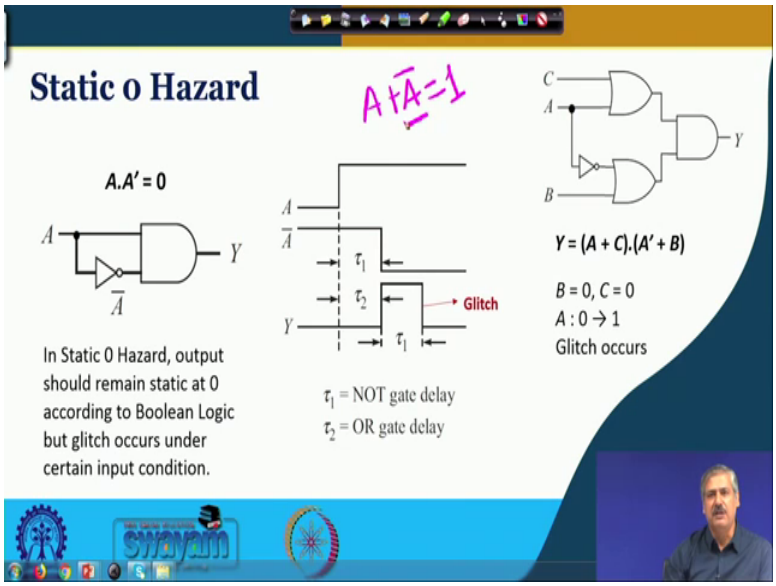
Lecture - 15
Static 0 Hazard and Dynamic Hazard

Hello everybody. In the last class, we discussed static 1 hazard. We saw that an otherwised optimized circuit can create issues in terms of producing glitches, which could be troublesome in some of the applications. And in that context, we found that how to detect such hazard for certain kind of you know circuits, and that is static 1 hazard and how to cover it.

(Refer Slide Time: 00:53)

So, in today's class we shall look at Static 0 and Dynamic Hazards. So, these are the two things that we shall take up today.

(Refer Slide Time: 00:58)



So, in static 0 hazard, we shall again the condition that we find is similar is analogous to what was there for static 1 hazard. In static 1 hazard if we remember, the Boolean expression reduced to something like A plus A prime equal to 1. Just to recollect the Boolean expression reduced to A plus A prime equal to 1, which always needed to be at 1, but because of the propagation delay involved in generating A prime through an inverter, we form A negative going pulse or a glitch occurring, and we found how to detect that and how to cover that.

So, in static 0 hazard, what is happening, the Boolean expression for a certain combinations of the input reduces to A an expression, which is something like A A prime, which should generate 0 as far as the logic of the circuit is concerned. But, because of the involvement of these time delay finite propagation delay in generating the NOT gate output, there is a glitch ok. So, this glitch will be A positive going plus positive going pulse ok. So, how does it occur?

So, if you look at the example, so this the input A is going from changing from 0 to 1 ok, and because of the input is input is going from 0 to 1. A bar is going from 1 to 0, but it takes a propagation delay, which is propagation delay for this inverter, this NOT gate going from high to low whatever time it requires, the finite propagation delay associated with the circuit whether it is a TTL base circuit or a CMOS base circuit. There is this delay associated is actually included in this duration time duration the tau 1 you see ok.

So, for this AND gate this particular time duration this 0 to tau 1 that one you see over here is the time when both the inputs are at logic high logic 1 right, so because of which whenever they are at 1 both of them are at 1, after the propagation delay involved for the (Refer Time: 03:36) this particular AND gate to go to logic high. So, it will go to high. And whenever it goes to low, it will go to low. So, during that period, it remains high. So, this is this positive going pulse that you see for a short duration because of the propagation delay involved here ok.

So, but for the AND gate you know I mean AND relation for this kind of relation you should always get a 0, which is not happening because of the finite propagation delay. So, in static 0 hazard, output should always remain static at 0 according to the Boolean logic, but glitch occurs because under certain input you know condition.

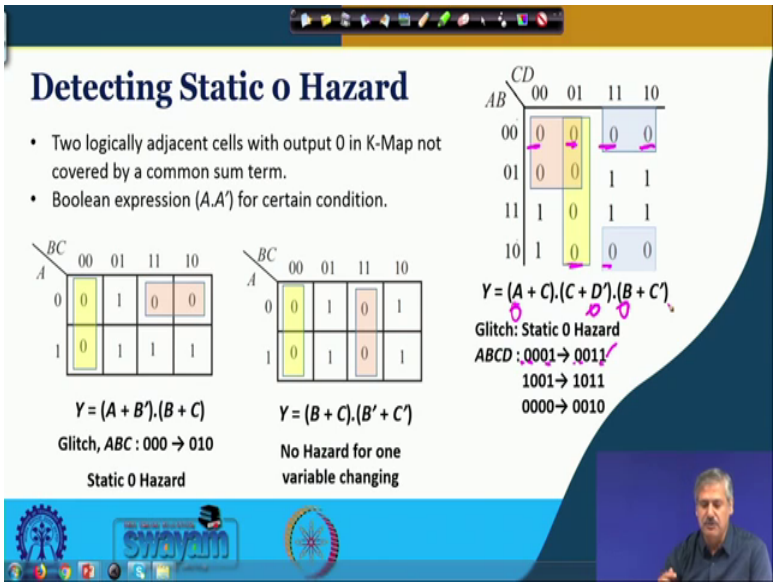
So, we look at one practical example. So, if you realise the truth table that we had seen before for the static 1 hazard problem, the same truth table if you try to realise through product of sum POS, then what will we get? We will get a relationship something like this. So, this is a known problem. This relationship we have seen earlier in the course ok, and to realise that we are having A or C generated here, and A prime B generated over here, and these two are ended through this AND gate ok.

Now, consider the input combination B is equal to 0, so, this is 0. C is equal to 0, so this is 0 ok, and A is going from 0 to 1. So, A initially is at 0 right, so which is making initially this is at 1 ok, so 0 and this is at 1, so this output is 0 right, this output is 1, but for the AND gate, we need both the output to be 1 to be out the for the output to be 0.

Now, this goes from 0 to 1 ok. So, this AND gate after the OR gate after the propagation delay go to goes to 1, but this one till this time this not gate output NOT gate output changes will remain at 1 ok. So, this output 1 goes to 0, after this propagation delay and this propagation delay ok, so because of which here it goes to 1 after one propagation delay, and it goes to 1 after two propagation delay if tau 1 and tau 2 are same.

So, momentarily for one propagation delay, both the inputs are at high which makes the output high, so because of which this glitch the one that we are talking about occurs in the circuit ok. So, for other input combinations, it will not occur, so that is what it is mentioned in the earlier class also that we take note of here that for a specific you know condition combination this thing the glitch occurs.

(Refer Slide Time: 06:42)



Now, how we detects static 0 hazard? So, static 0 hazard detection is similar to static 1 hazard detection ok. So, two logically adjacent cells with output 0 in the K-map, if it is not covered by common sum term ok. In static 1 hazard, it was two logically adjacent cell with output one not covered by common product term ok. So, in this case it is the output having 0, which goes into the making of you know previous uh expression, they are not covered by common sum term.

So, from the Karnaugh-map, we can find this one. And if you look at the Boolean expression, which ultimately reduces to A A prime kind of thing A variable and it is compliment ended together for certain input combination ok. So, we do not need to go up to the expression level some simply by looking at the Karnaugh-map grouping from which the expressions are generated, we can detect presence of a static 0 hazard.

So, we look at one example. So, this is an example, where you see this 0, this is this is optimized expression all right. So, these two 0's and these two 0's, they are grouped together. And A plus B prime is one term this term, and B plus C is the other term and they are ended together ok. So, there is a potential glitch static 0 hazard why, because this 0 and this 0, they are logically adjacent ok. And they are not covered by one common sum term, is it clear?

So, how when does the glitch occurs, when A is equal to 0, and C is equal to 0 A is equal to 0, C is equal to 0. So, this is the two cases, when A is equal to 0 and C is equal to 0,

and B changes from 0 to 1, then the glitch occurs going by whatever we had just discussed the examples; so, similar example over here. Now, if you look at the other truth table a similar truth table, in relation two what we discussed in the static 1 hazard. So, there we grouped the 1s, now we are grouping the 0's, so this is one group, and this is one group. And there generating term B plus C, and B prime plus C prime, and we are ending them together to get the POS representation.

Now, we see that the logically adjacent 0's these two, they are under common sum term these are two logically adjacent 0's there are under common sum term, and these 0's are not logically adjacent. So, they need not be covered by common sum term ok. So, we do not have any hazard here, we do not have any static 0 hazard here ok, so that is what we take from this particular truth table, and converted to Karnaugh-map ok.

Now, we look at a four variable example. So, this particular example, you see the presence of you know 0's and 1's in that truth table the way it has been mentioned. And to get the first the optimized expression, so we look for grouping of 0's with largest you know possible group size. So, this is one group, this is one group right, which is generating this term A plus C right. Then we have this particular group all four 0's, they can be grouped together right. So, this is generating C plus D prime.

And the other two 0's, these two 0's, and these two 0's, they can be part of one group, group of four members, so that is between them 0 0. So, B plus C is remaining with one, so C prime. So, this is the third term that we are getting, which is covering all the 0's right. So, this is the optimized expression. But, does it have a you know static 0 hazard.

So, for that what we need to examine, we need to see whether the logically adjacent 0's are covered by common sum terms or not. These are logically adjacent 0's covered by common sum term, but here is one logically adjacent 0 I mean 0's one pair, they are not covered by common sum term ok. So, this might lead to a glitch for a specific input combination right.

So, is there only one possible way the glitch can occur no, you can see that there is another such thing ok, which is there you know logically adjacent, and but they are not covered by you know common sum term is that the on you know second on possible thing there is there no other ways glitch can occur in this particular example. There is another way glitch may occur if you look at this 0, and this 0 ok, it is they are also

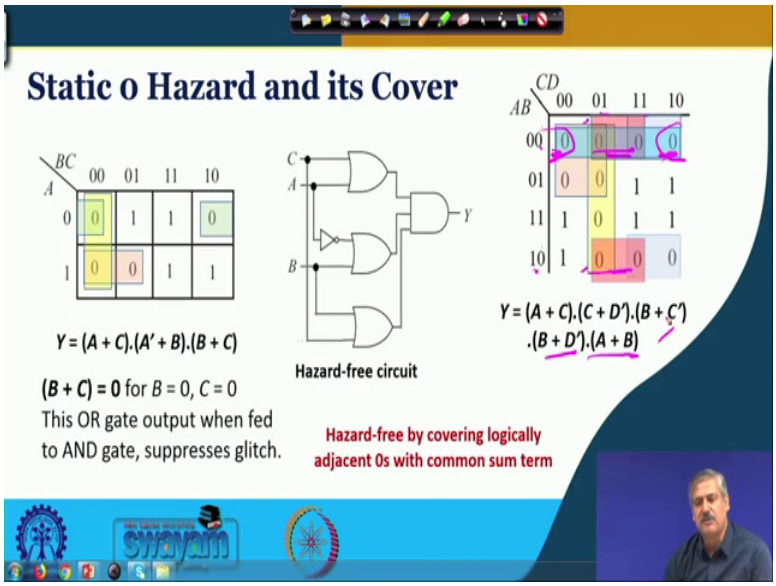
logically adjacent. So, and there is no common term common sum term covering them ok. So, there can be a glitch for that also right.

So, what are the in this possibilities the combinations? So, between these two you can see AB and AB remaining 0, D remaining 1, and C changing from 0 to 1. So, A remaining 0, B remaining 0, D remaining 1, and C changing from 0 to 1. So, A 0, B 0, D 1, C changing from 0 to 1 ok. So, this is a case where ABD are 0 0 1, C changing from 0 to 1 a glitch a negative a positive going pulse will occur, when it was logically it is supposed to be at logic 0.

If you look at the expression, if you substitute we can see that the same thing will be happening. So, A is equal to 0, so this is 0 right, then B is equal to 0 B is equal to 0, this is 0, and D is equal to 1 right. So, D is equal to 1 means D prime is 0 right, so what we will get C and C and C prime, so C C prime is occurring right a variable and its compliment. So, this is a case, where there is a possible a you know glitch ok, a glitch will occur if we make the make a transition if the input C make a transition from 0 to 1.

Similarly, you can check for the other cases also right. So, this is the case when 1 is 1 right, b is 0 right and d is 1, d is 1 between these two right. So, A 1, B 0, and D is 1, and C is changing a transition from 0 to 1 glitch occurs. And the last one the 3rd one that we talked about these two, so A is 0, B is 0, D is also 0 right between these two 0's and C changes from 0 to 1 C changes from 0 to 1 ok. So, there is a glitch occurring right. So, this is the we can figure out three such you know possible glitches, and the this particular realization this particular realization has a has static 0 hazard ok.

(Refer Slide Time: 14:42)



Now, to cover we do not we have detected, but we need to think of how to cover it. And again we follow the logic that we followed for static one hazard covering. So, the one that causes the a specific hazard, we need to make sure that that cost does not arise. So, for that what we will do simply, we shall see that the adjacent logically adjacent 0's are covered by one common sum term.

So, in the previous example a in this particular example right, what we are looking at this two 0's right, we are covering by a common sum term ok. So, this is the example with which actually we started. So, this particular circuit is of this is one sum term A plus C right A plus C ok. And this is another sum term, which is A prime plus B A plus C and A prime plus B right.

So, this is the one with which we started today's discussion right, so if we quickly go back. So, this is the circuit we started with right, and we found that there is a possibility of a glitch. So, we are looking at covering this particular hazard ok. So, these are the two terms right. And then the common term over here is B and C, so we are forming a group of them, so group of it, so B plus C.

So, when we are just forming this term, this is so called redundant term in terms you know optimization problem, but together it solves the problem, I mean the addresses the static 0 hazard. And what how does the circuit changes, so basically this is the B plus C that comes over here. And when B and C are 0, these two are 0, this output is 0. And this

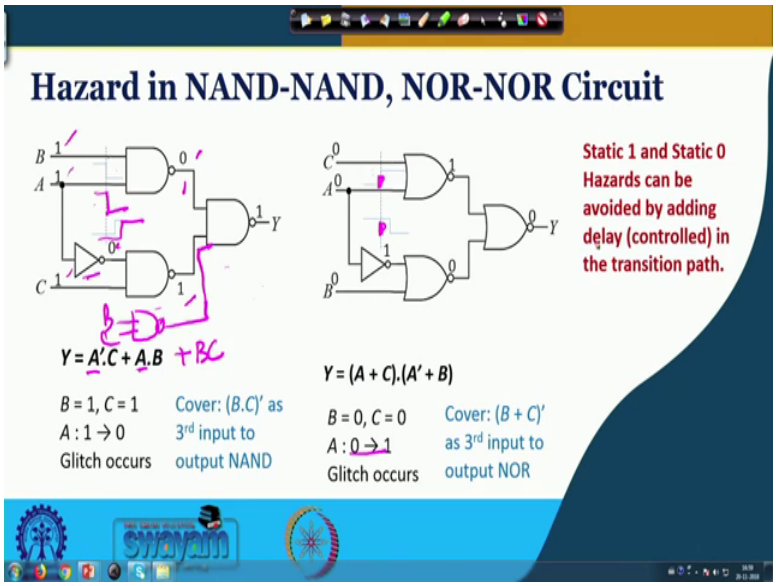
holds, this is AND gate input. So, if there is a 0, it will be always a 0 right. And even if some changes are occurring over here A going from 0 to 1, that is not go going to the final output ok. So, this is similar to what we had seen in case of static 1 hazard a 3rd AND gate was there, which was feeding to the final OR gate ok. So, a just analogous the in terms of POS representation, is it clear.

So, now we look at the other example the for variable example. And there are three glitches three possible you know glitches that were was to be covered that were to be covered. And so in this case what we are finding, we find that to cover this one and this one, I mean these two 0's these two 0's, we can form a larger group is not it. We can form a larger group I mean group of 0, otherwise you could have one term over here, another term one common sum term right.

So, then we were would have required three two such terms with three literals, so but we know that we can have a larger group of four, which will make sure that all adjacent 0's like these two, and these two are under one common sum term. And for that what we are having? So, in this B is remaining constant, and D is remaining constant, B is remaining constant with 0, and D is remaining constant with 1. So, B plus D prime is the term that will be required ok.

And the other thing is these 0 and these 0 that was that were to be put under one common sum term ok. So, one option is to have you know like this, but again that will require three literals literal cost would increase. And what we would like to do to avoid that we will be having a larger group like this with only two variable invert, so that is A plus B. So, because AB is remaining constant over four this four 0's ok. So, with all of them together we can see that the static 0 hazard for the circuit all the three possible cases are covered right.

(Refer Slide Time: 19:36)



So far we had seen the static 0, static 1 hazard and static 0 hazard a using AND, OR, OR AND kind of you know two level circuit, we are only talking about two level circuit here as we have noted before. So, if we had used instead you know NAND-NAND or NOR-NOR for realization of these two level circuit ok, what would have been the case ok. We may have certain curiosity about it.

So, let us look at one quick example. So, if this is a realization of A prime C plus A B ok. So, this two level realization in this case is achieved by using only NAND gates ok. So, these are the two NAND gates. And finally, this is the third NAND gate, which is just combining the output of the previous two NAND gates ok. So, this is we are familiar with from our week-2 discussions how we are getting this kind of circuits. We can use De Morgan's theorem also on AND, OR, and then we can get NAND-NAND kind of circuit ok.

So, in this case what happens, so we had seen before, so we continue from there that A bar plus A this comes when B and C are 1, is not it; B and C are 1, A and A bar A plus A bar kind of relationship is coming. So, that is as A you know potential glitch and when A goes from 1 to 0 right that is the static the glitch will occur right

So, let so what is happening over here this is NAND-NAND earlier we had studied AND, OR. So, in this case initially let us considered all the situation B B is 1, C is 1, and A is at 1 ok. So, A is at 1 means this NAND gate NOT gate output is 0 which is fine. So,

what are the corresponding NAND gate outputs. So, this is 0, 1, 1, this is 0, and this is 0 and 1. NAND gate 0 is the forcing input. So, this is the output is 1 ok. So, 0 and 1, output is 1 which is fine

Now, this is the thing that you see that at this point of time at this point of time NAND gate is making this input A is making a transition A to 1 to 0 right. So, this has become 0 after 1 you know delay will become 1 right. So, but here this 0 to 1, this 0 to 1 that you can see goes after 1 propagation delay over here. So, it will not when it is becoming 1, it is not immediately you know going to 0, because input 2 is fail is delayed I mean the input of 1 is delayed by the propagation delay of the NOT gate. So, for one such delay period of this not gate both of them are 1 both, this one is 1 this one is also 1. So, this 1 1 will be make the output 0 for that amount of time including involving all the propagation delay of this gate and this gate right, so that is the glitch we are talking about. So, this similar glitch occurs the one kind of thing that we had seen for that relation it is occurring over here also ok.

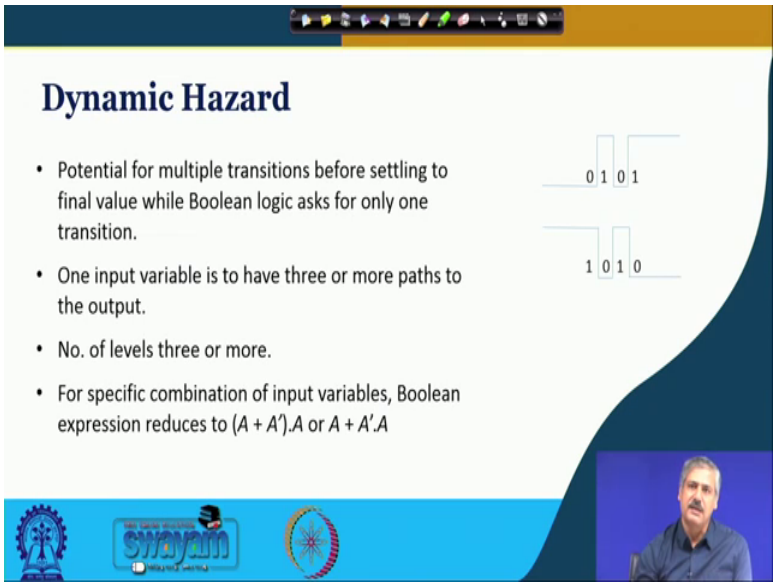
So, how to cover it? So, to cover, so this B and BC will be the 1 that we had seen we earlier. We this is the one that we had considered one AND gate and which was going to the third input of the OR gate similarly it will be B C, NAND, one NAND gate will be there with B and C right, and this will go as the third input over here ok. Similarly, for POS realization right, if we go for NOR-NOR right, we shall have B is equal to 0, C is equal to 0, this is the you know snapshot of the different logic values. Then A goes from 0 to 1 ok.

So, A goes from 0 to 1 over here, and then A bar goes from 1 to 0 like this with this propagation delay. So, momentarily both of them are having so this is having 1 sorry and this is also having 1 right. So, for a NOR gate 1 is the forcing input. So, this is 0 at that time this is also 0. So, 0 0 the output is becoming 1 ok. So, this is something that we take a note of that whether you realize by NAND-NAND or NOR-NOR or the corresponding AND or OR are the result remains the same and the covering is a similar.

And one might note that if we can introduce in this path it delay which is which is you know similar to this delay, then this can be avoided. Similarly, if you can add you know a buffer which is providing this amount of delay ok, but that is we only to be careful that this delay is added in appropriate amount; otherwise it will not help the situations. So,

better to cover in this manner, but by adding delays also we can avoid it.

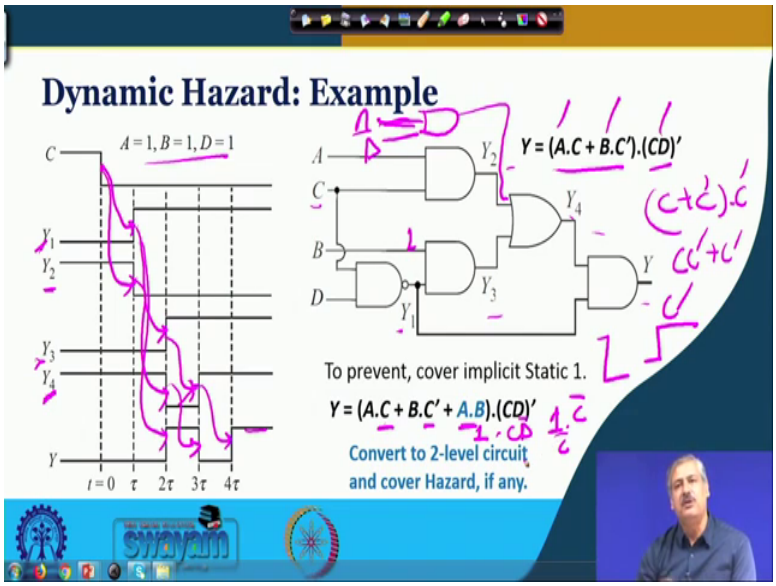
(Refer Slide Time: 25:22)



So, dynamic hazard we shall discuss now. So, dynamic hazard is the case when there is a potential for multiple transitions ok, while this circuit is required to make one transition from 0 to 1 or 1 to 0, but it is making more than one transition. So, this is what is known as dynamic hazards. So, basically is the changes the way we have seen in the right hand side, so this kind of you know changes take place.

And for this one notice that one of the input variable which is actually causing this thing is to have three or more paths to the output, and number of levels involved earlier we saw the there are two levels. So, number of level should be three or more ok. And inherently one can see the relationship like what you see over here A plus A prime and A or A plus A prime and A this kind of thing so A in A inherently there is a static 1 and static 0 kind of relationship emerging somewhere, and this is what is visible in a dynamic hazard causing circuit ok.

(Refer Slide Time: 26:38)



So, we shall we will look at one example is a example where you can see this is AC plus BC prime right ended with CD prime. And we are looking at a situation when A is equal to 1, B is equal to 1, and D is equal to 1. So, potential you can see C plus C prime and C prime this kind of you know situation emerge ok, so this just example. So, what happens at that time you can see from this particular diagram, so this is the value of the other inputs ok. And C is making a change, these are the outputs Y 1, Y 2, Y 3, Y 4 and Y intermediate outputs for the purpose of plotting. So, C makes a change a makes a you know transition from 1 to 0 right. And then because of that Y 1 will go to go from 0 to 1 because D is held at 1 right, so then it will go to 1 after one propagation delay right.

What happens to Y 2, Y 2 is and gate right. So, if input changes from 1 to 0, so the output will become also 0 because C was 1 C was held at 1. So, output was 1 because of presence of A has 1. So, because of which this Y 2 is also becoming 0 after one propagation delay. We are considering same logic family more or less same you know propagation delay term ok. After that it remain same at that value. What happens to Y 3, Y 3 is AND of B which is 1, and AND of Y 1. So, both of them when it is 1 the output will become 1. So, both of them are 1, so this is Y 1 is 1 input right and B is already 1. So, it goes to 1 over here. So, it will take time one propagation delay for this and gate right for which Y 3 becomes 1 over here.

Now, what happens to Y 4? So, Y four is or of Y 2 and Y 3 right any one of them high

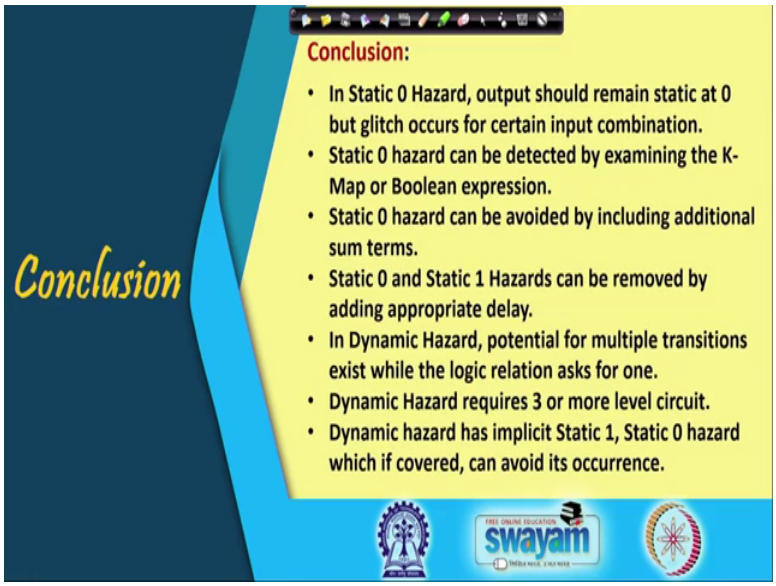
output will be high that is the idea. So, this is the Y 4 we are talking about. So, this Y 4 if we look at we are we need to look at Y 2 and Y 3 right. When any one of them is 1 the output will be 1. So, you see that Y 2 is 1 here this is low, but Y 2 is becoming low in this place. So, after one propagation delay, this will become low right. And Y 3 is going high here in this place. So, after one propagation delay this will become high right.

Finally, Y 4 and Y 1 ended together is the Y right. So, this is Y four and Y 1 ok. So, this is the Y 1 both of them need to be high. So, Y 4 was 1 right. So, this one making it high after one propagation delay right both of them need to be high Y right. So, after that this Y 4 has gone low here, so that is making low in this place right, then Y 4 goes high. So, this is go is going high ok. So, then what you see the Y is like this.

And if you look at the corresponding logic here C plus C prime and C prime, so basically it is CC prime plus C prime, so C prime only. So, when C changes from 1 to 0, output should go from 0 to 1, the but here output is going from 0 to 1 then again coming to 0 and then again going to 1. So, instead of one transition, it is making multiple transition. So, this is what is seen as dynamic hazard and how to cover it, it is relatively more difficult than static 0 and static 1 hazard.

We have to see what is the this inherent static 1 hazard can be seen over here because of the C plus C prime kind of relationship, and to cover it we will be having it is happening for A is equal to 1 B is equal to 1 case which we have seen investigated before. So, you add one additional term right AB without I mean making any difference in the truth table as such this is the redundant term, so that will be coming in parallel to this as A B A and B right, and that will be getting connected over here right. And after that we will be seeing that when your A B and D is equal to 1 right, so this is at 1, AB is held at 1, so 1 and CD prime. So, basically 1 and C prime that is what we will get or C prime ok, so only one transition will take place.

(Refer Slide Time: 32:01)



So, with this we end today's discussion very quickly. Summarising what we discussed today static 0 hazard output should remain static at 0, but glitch occurs for certain input combination. Static 0 hazard can be detected by examining the Karnaugh map and the or the Boolean expression. It can be avoided also by including additional sum terms and appropriate delay I mean that we have noted. In dynamic hazard, potential for multiple transition exist and instead of you know only one which is required from the logic relation. And it requires 3 or more level of circuit. And the inherent implicit static 1, static 0 hazard if identifying identification of that and covering can avoid its occurrence.

Thank you.