Architectural Design of Digital Integrated Circuits

Prof. Indranil Hatai
School of VLSI Technology
Indian Institute of Engineering Science and Technology, Shibpur, Howrah

Lecture - 19
Efficient Adder Architecture (Contd.)

Welcome back to the course on Architectural Design of ICs. So, we were discussing in the last classes; that means, the Adder Architecture. So, we have started with one that binary addition operation. Then, we have discussed that means, for n bit addition where are the adders are; that means, they are a different adder architecture, so, different adder architectures for different aspects or different application. So, different application means somewhere I need high speed, somewhere my requirement should be low power consumption and some somewhere it might be for low area consumption.

So, we have seen that there are several type of adder architecture which are available; that means, people have tried for to design different, though the addition operation it do that means, that keeping the functionality same how the architectural. That means, changes basically whenever we are doing the VLSI implementation, at the time in the hardware architecture changes, how it effects the it is major constraint that speed power area that we have we are discussing in the last few classes about that there are different adder architecture like, this we have started with ripple carry adder, then we have seen that carry skip adder.

So, in the carry skip adder, we have seen that it is, if we of choose the variable sized carry skip adder, then it we are getting; that means, the delay is optimized, then if we use multi level variable blocker. That means, this carry skip adder or block adder, then more the delay we can say that means, more the speed or the operating frequency you can achieve.

And then, we have seen that we will carry look ahead adder. So, this ripple carry adder and carry look adder, these are the two basic adders we have already known from the basic electronics of circuit design. But this is; that means, from the ripple carry adder as the carry is rippling from LSB to the MSB side. So, for if the number of bit; that means, the addition if I require the number of bits is more at the time, the carry is more, sorry the propagation delay for the carry signal is more.

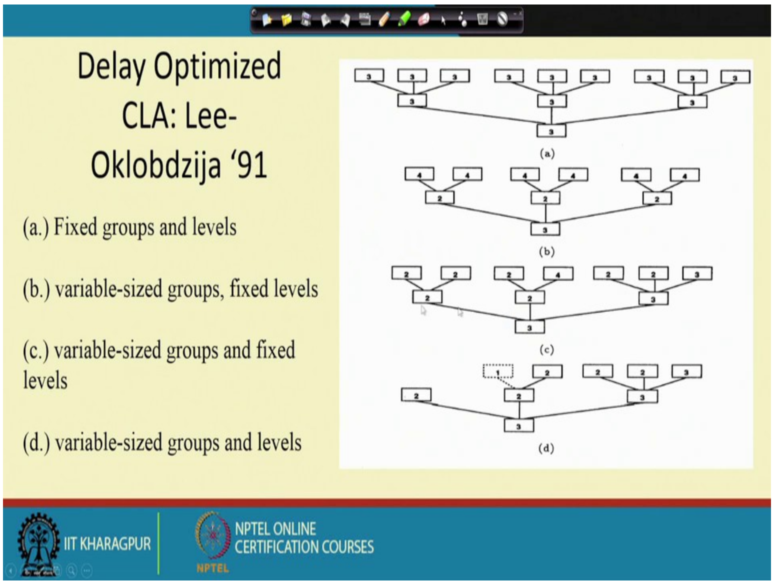
So, to reduce that, we have designed what we have, we have design or we have come to the point that if we can generate, we do not we will not wait for the carry signal for; that means, rippling from LSB to MSB side. So, instead, what we can do? We can generate and in each level we have; that means, we have observed when we have seen that the carry generation and carry propagation logic is there. So, identical; that means, initially, we can based on the input bits. So, we can generate and propagate the carry in such a way so that the corresponding delay that can be reduced in look carry look ahead adder ok. So, that we have seen.

And whenever we are, we are doing this carry look ahead adder design. So, at the time we also have seen that though this carry skip adder or this look ahead carry adder, what we are doing? We are basically; that means, that carry is basically traversing or that is that the critical path which is follows in the like horizontal direction. But in case of a look ahead carry adder, the; that means, the carry or the; that means, or the carry or the; that means, the propagation delay that follows in the vertical direction. So, why vertical direction; basically, that also we have seen because this c 0, that is this g 0 plus p into c i; that means, c i plus 1 that is equals to g i plus p i into c i.

So, whenever the c i means that is the previous carry which is multiplied with this propagated one; that means, p i is nothing but a XOR b and g equals to a AND b; that means, the g is the, that means, generate the carry which is the logic for which is a AND b and for propagating the carry, the logic is a XOR b. So, that has been and at this propagated carry, that has been ended with carry previous carry and now, then we are getting if we are just OR with this generated carry then we will get the next carry ok. So, like that way, we just do that ok, that we have already discussed.

So, here today, we will see another things. In the last class also, we have discussed that there is that fan in and fan out dependency. So, what is that fan in, fan and fan out dependency? And that we will see now. So now, we will see this Delay Optimized CLA; CLA means look ahead carry adder carry look ahead adder.

(Refer Slide Time: 05:29)



Delay Optimized CLA: Lee-Oklobdzija '91

(a.) Fixed groups and levels

(b.) variable-sized groups, fixed levels

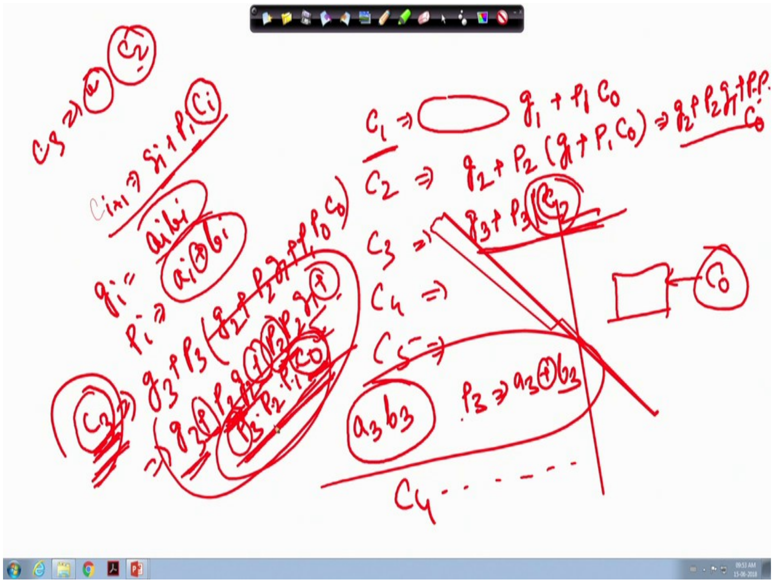
(c.) variable-sized groups and fixed levels

(d.) variable-sized groups and levels

So, if you see, this is basically the work follows of Lee and Oklobdzija in 91. So, here you see there are four different architectures which follows this CLA method.

So, here you see, the first one says that the Fixed groups and levels. So, here you see, each of the that means, whenever we are, that means, generating the corresponding carry ok, so, the as we if we just.

(Refer Slide Time: 06:06)



$C_1 \Rightarrow g_1 + P_1 C_0$
$C_2 \Rightarrow g_2 + P_2 (g_1 + P_1 C_0) \Rightarrow g_2 + P_2 g_1 + P_2 P_1 C_0$
$C_3 \Rightarrow g_3 + P_3 (C_2)$
$C_4 \Rightarrow$
$C_5 \Rightarrow$
$P_3 \Rightarrow a_3 \oplus b_3$
$a_3 b_3$
$C_4 \cdots$
C_0
$C_3 \Rightarrow C_2$
$C_{i+1} \Rightarrow g_i + P_i C_i$
$g_i = a_i b_i$
$P_i \Rightarrow a_i \oplus b_i$
$g_3 + P_3 (g_2 + P_2 g_1 + P_2 P_1 P_0 C_0)$

So, the carry signal the carry signal, c i plus 1, that is basically g i plus p i into c i. So, this c i is the previous carry and g i and p i, logic for g i equals to a i b i. And for p i, the logic is a i XOR b i.

So; that means, now, now if I just whenever I will continue, with this particular equation, so, at the time for c 1, I will get some of the expression over here in corresponds to a i b i, then c 2, I have to find out, then c 3, then c 4, then c 5 ok. So, something like this, but whenever we will see suppose for c 1, what will be the case if I want to c i will be this g g 1 plus p 1 into c 0 ok. So, then c 2, what will be c 2? C 2 will be g 2 plus p 2 into g 1 plus p 1 c 0. So; that means, now this is g 2 plus p 2 g 1 plus p 2 p 1 and c 0.

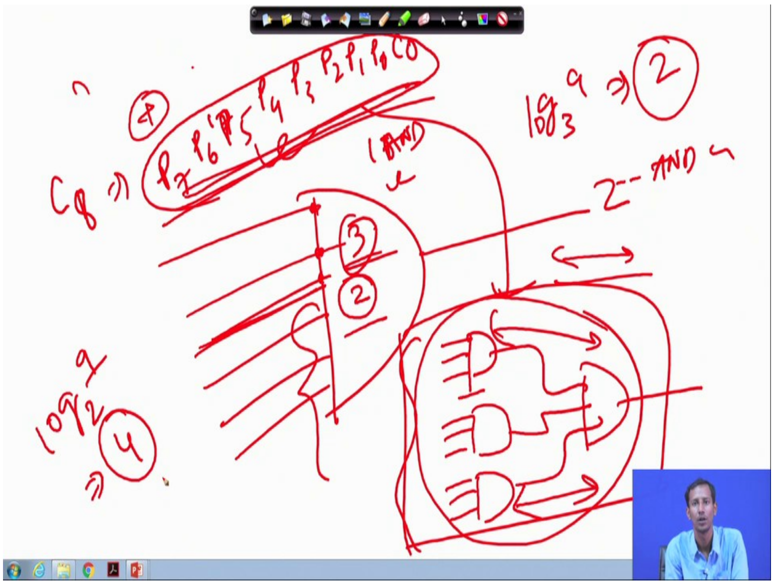
Then next, the next what will happen? In next, again if I want to calculate c 3, then it will be g 3 plus p 3 into this sorry p 3 into c 2. So, now, c 2 is this value. So, if I just; that means, replace this c 2 by this value, so, at that time c 3 will became g 3 plus p 3 into this g 2 plus p 2 g 1 plus p 1 p 0 and c 0. So; that means, g 3 plus p 3 g 2 plus p 3 p 2 g 1 plus p 3 p 2 p 1 and c 0. Here you see, whenever we are just; that means, this generating or we are getting the carry, which is basically depends on only the first carry input bit which what we basically know from the beginning what is the carry last. That means, carry in bit to the corresponding that is 4 bit CLA or 8 bit CLA. So, what will be that this c 0 that we know from the beginning.

And what about this g 3, p 3 this and this p 2, g 1 this; so if I just want to; that means, if I just want to calculate this, so, how can I calculate? Basically, g 3 is nothing but a 3 and b 3. So, this depends on the current input and this g 2 sorry if I just want to calculate this p 3, so, p 3 is nothing but a 3 XOR b 3. So; that means, it does not depend this carry that does not depend on the previous carry which happens in the carry look adder sorry ripple carry adder.

So, in ripple carry adder, the c 3 cannot be calc computed unless and until this c 2 is available to me. And c 2 is again that cannot be calculated unless c 1 is available to me; that means, once the c 1 competition of c 1 is; that means, over then only I can calculate c 2 so; that means I have to wait for that time. But, here I do not have to wait because c 0 is already available and all these signals all are available at that time only.

So; that means, that and by that we basically increase or we decrease the propagation delay in carry look adder but at what cost? What we are doing here? We are just increasing this level. So, as I am considering this. So, you considers for c 3 I am getting this p 3 p 2, p 1, c 0. So, for then again for c 3 this level again it will increase; that means, here I need one, two, three, four, so; that means, this 4 of; that means, or operation I need you do then one, two, three, four, so, three of this AND operation I, I have to do for this particular thing; that means, whenever we will increase, what does, what actually I want to say that is.

(Refer Slide Time: 11:21)



If I increase this particular c for c 8, so, at that time for c suppose for c 8, so, at that time it will be something like this p 7, p 6, p 5 sorry, p 5, p 4, p 3, p 2, p 1, p 0 and c 0. So, this level that will be the last term

So, it will increase something like this. So, at that time, so, to implement this, what I need what I can do I need a 1, 2, 3, 4, 5, 6, 7, 8, 9. So, nine input and gate or else what I can do? If I am having this that is why basically we are. That means, we I said that there is fan in and fan out dependency, because this gate the fan in, if fan in is maximum of fan in is 3. So, that means, it can take only three input bit, it cannot take nine input bits ok. So, at the time if I just want to; that means, do this ANDing operation using 3. So, at the time what will happen? So, 3 then again 3, then again 3, so, then, this 3

So, at that time this will be the circuit for implementing this so; that means, here what I am doing, I am just increasing the level ok. So, the logic level has been increased by here it was just only one gate delay, one AND gate delay but here it will be two AND gate delay.

Now, if I say is that fan in is maximum it is 2. So, if it is 2, then again it will increase. So, at the time; that means, the level will be increased so; that means, how the level will be increased that it will be log 2 base n. So, this means 16. So, 4 level I have to do. So, whenever we are considering this has 3. So, log 3 base 9, that is equals to 2.

So that means, for these particular terms and there will be again for this OR operation will be there for this, if I just go back to this slide. So, for these particular terms will also increase, so that means, this OR operation will also increase. So, or operation will increase means, again I need this OR gate for the final; that means, ORing operation. So, I need one OR gate of 8 or 9 input. So, again if I say that the thing is that; that means, we have the fan in limitation of 2 or 3. So, at the time again the level will be increased something like this ok.

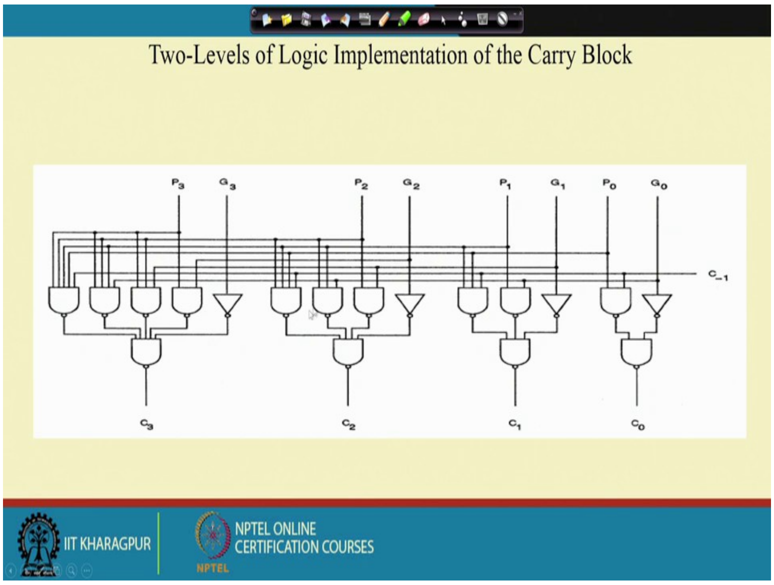
So, that is why how we can basically, how we can optimize the CLA; that is there are if we consider different kind of topology whenever we are implementing this for carry input; that means, the carry bit, so that; that means, for different topology, we will get different numbers in the corresponding critical path.

So, here you see actually for fixed groups and levels so; that means, this fixed groups and levels means, here I need what; that means, the input corresponds to the input corresponds to the gate is 3, then total 9 bit, again that has been ANDed or OR with this; that means, this is one structure fixed groups and levels means where each of these follows the same number ok.

So, then next is that variable sized groups, but 6 levels, but here you see, at the first level there are 4, in the second level there are all 2, in the third level it is 3 so; that means, we are considering this 4 input gate at the very first level and whenever we are at the second level at that time we are following thus; that means, the level or the fan in of 2 and in the level 3, we are considering fan in of 3. Then another; that means, kind of topology will be that there; that means variable size groups and fixed levels. So, here the levels are fixed, but the groups are variable. So, there 4, then 2, then 3, but here the variable sized groups and fixed levels, so, here you see; that means, this number will be now different. So, here you see this is 2, 2; then here 2, 4; then 2, 2; then here 3 so; that means, the groups are different here I am having 2, here I am having 3. So, something like this; that means, the groups are different, but the levels are fixed which has 2 in this case sorry 2 in this particular case. So, 3 for this case 3 for this case.

Then variable sized groups and levels. So, here you see in this particular architecture, both the groups and the levels both are basically variable. So that means, in different of this we will get different delay for the same logic implementation.

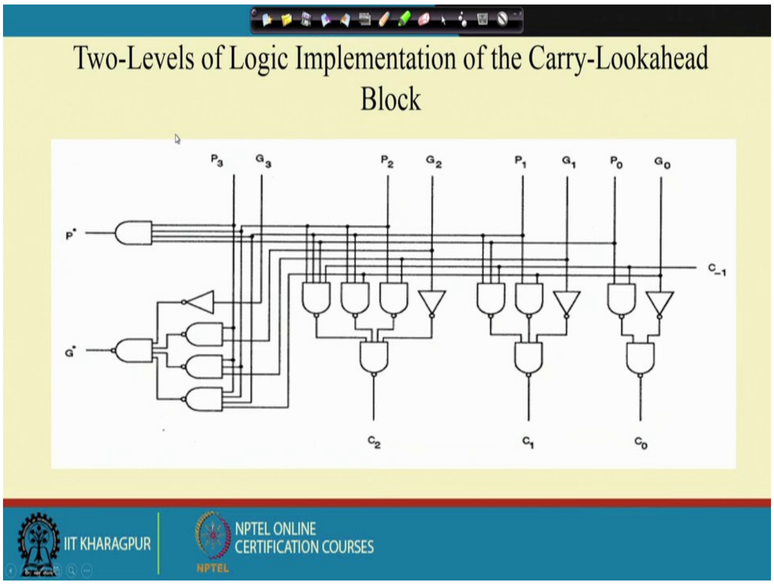
(Refer Slide Time: 17:48)



Two-Levels of Logic Implementation of the Carry Block

Then, this is the two level of logic implementation of the carry block; that means, you see there are only two a number of gate; that means, this is the inputs. So, there are only two; that means, from input to the output there are only two number of gate delay in each of these particulars things ok. So, here you see, here for this the input are 2, for this this is 3, for this this is 4, this is 5 for this this is 5, then 4, then 3, then 2, then 1.

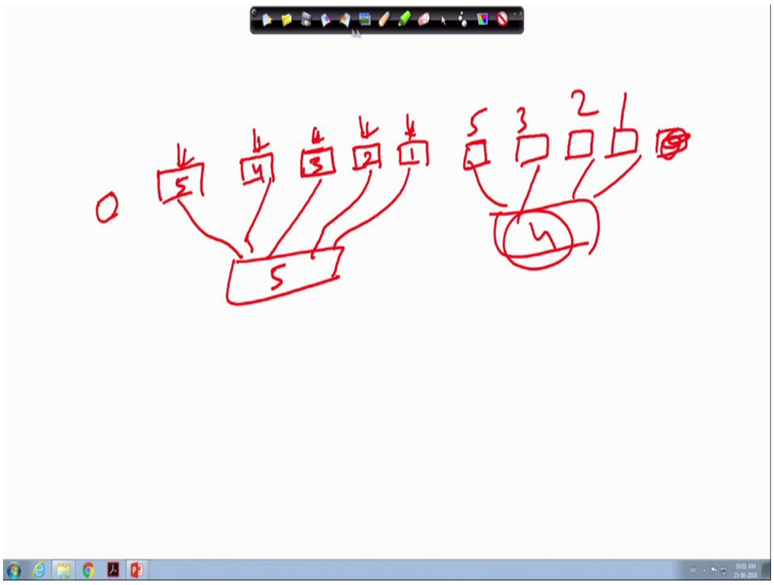
(Refer Slide Time: 18:28)



Two-Levels of Logic Implementation of the Carry-Lookahead Block

That means, now if I just; that means, go back to the previous slide, at the time this says that this is if I just corresponds to this particular circuit ok, so, at the time what will happen here 5, 4, 3, 2, 1 and here 5, then 5, 3, 2, 1, then here 4. So, something like; that means, if I just draw it.

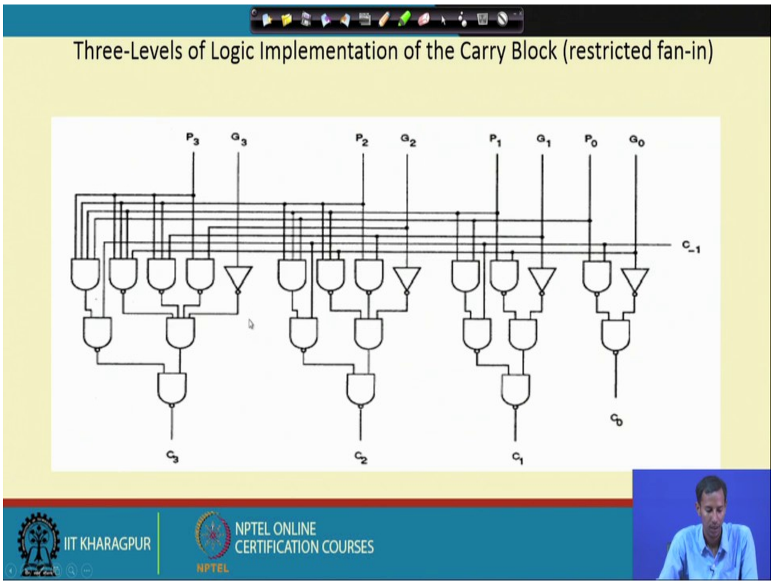
(Refer Slide Time: 18:55)



So, at the time how it will be look like 5, then 4, then 3, then 2, then 1. Then, this is our 5 then again sorry then 5, 3, 2, 1 ok. So, something like this; that means, it has 5 input, it has 4 input it has three input, it has two input, it has one input ok.

Then again, whenever we are considering this particular if I just, so, this is the two levels; the levels are two only; that means 1 and 2. Then, next you see two level of logic implementation of the carry look ahead adder block; that means, this is another implementation technique. So, here you see instead of this vertical, if I just put it in the vertical, it will be there that is the same.

(Refer Slide Time: 20:15)



Three-Levels of Logic Implementation of the Carry Block (restricted fan-in)

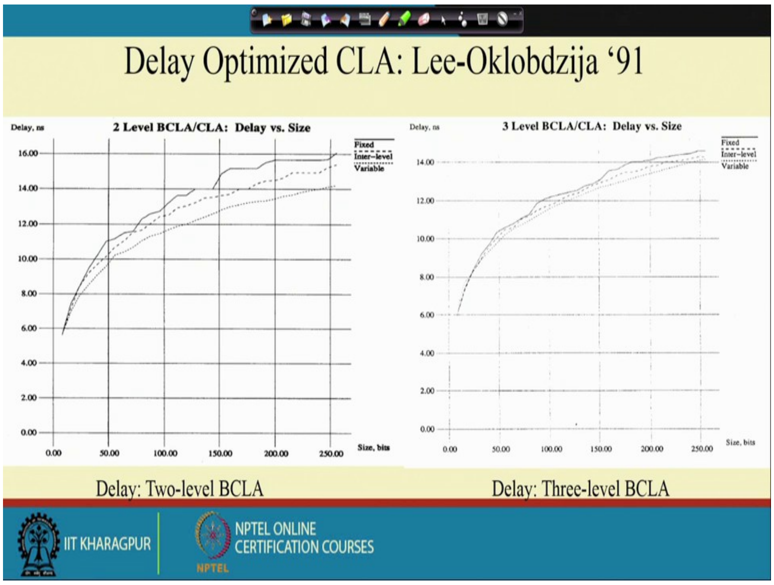
And then, three level of logic implementation of the carry block and with restricted fan in. So, restricted fan in means, if I just; that means, restrict that the minimum number of the fan in will be 3 or 4. So; that means, it cannot go beyond 4. So, at that time what will happen, the same circuit, the same circuit this in this case there was five input and the five input. If I restrict the fan in of 4; so at that time at that time what will happen I, I need one AND gate which is basically doing this 4, then again I need another one, where I will do this fifth number, I will just consider. That means, initially it was considering only one, one gate, but here it is considering two gates, you just see here I am considering only one AND gate of five input. But here, I am considering one AND gate, then one AND gate; that means, one our input and gate then one two input and gate ok.

And then, again if you just see that another changes, here there was five inputs. Again I said that that for the gate the restrict fan in is restricted for 4. So, if I just come to this then again this is ANDed first in 4 then again it is and ANDed with this and this to get this; that means, that 4 that now it has splitted into two levels

So, this is that means, for this if I just restrict to 4, for this if I just restrict to if I can restrict to 3, in this particular case if I restrict the corresponding to for 4, for this also if I just restrict the term to 2; so; that means, here in this particular case, the fan in is restricted to 4 in this particular case the fan in is restricted to 3, in this case the fan in is restricted to 2 from the circuit basically I am telling. If you just go back then you can see here we are having 5 input; if I just restrict for these particular things the fan in of 4. So, at the time this circuit is very much I do not need to change, when I need to change, when if I restrict these to 3. So, at the time I need two levels here, two levels here.

Here also if I just restrict to 3, so, at the time it does not; that means, it does not need any change, but whenever I will restrict the fan in to 2, 2 specifically 2, 2. So, at the time this will change like this. So, this is the three levels. That means, whenever we are putting this that means we are increasing. If we restrict the fan in; that means, we are increasing the corresponding levels by 1

(Refer Slide Time: 23:50)



Delay Optimized CLA: Lee-Oklobdzija ‘91

Delay: Two-level BCLA

Delay: Three-level BCLA

IIT KHARAGPUR

NPTEL ONLINE CERTIFICATION COURSES

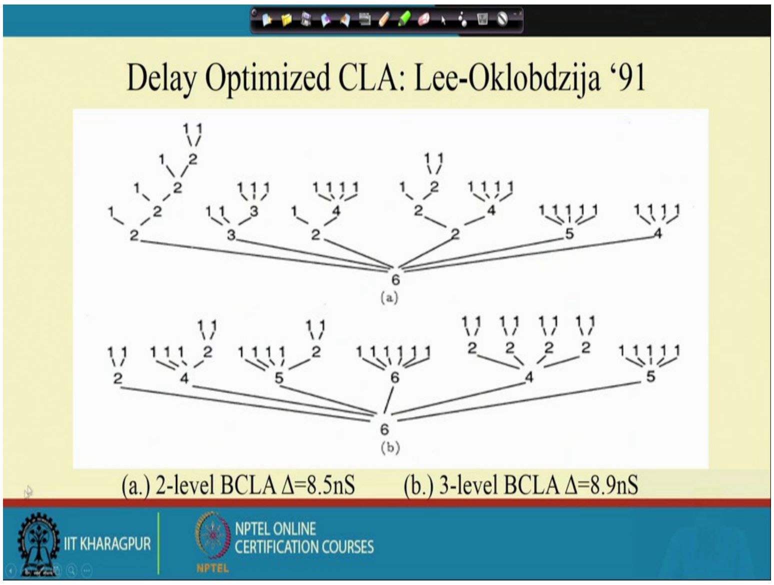
So, this is again this is another implementation of this carry look ahead adder with restricted fan in. And if you just see the delay optimized this CLA, the performance of that. So, at the time, you see this; that means, the straight line is the first one is the and this is the basically delay and in these this size this is the number of bits or the size we; that means, that number of n if I consider.

So, if you see that for fixed levels CLA ok, for fixed level CLA, the delay are following this particular curve. And if I follow if this dotted one these are the variables; that means, you that it can be in that variable size or groups or fixed groups, variable size or fixed size, sorry fixed groups, variable levels ok. So, four of the architecture we have seen; that means, four of the topology we have seen, so, any from that that is a variable 1. So, that in that case it follows a lesser number of delay. This is for two level of; that means, CLA design and if I follow three level of CLA design. So, at the time also you see this variable; that means level CLA that gives us better result in terms of this; that means, critical path reduction ok.

So, why critical path reduction, that means, if I consider more number of; that means, if I consider; that means, the fan in of 9. So, at the time because of the loading effect or because of the parasitic which are related or which are associated with the corresponding gate design at the transistor level that becomes too much high which basically increases the delay. But here whenever we are considering lesser fan in gates; that means, whether that is 2 or 3, so, at that time I will get more number of savings in terms of critical path or in terms of delay ok.

So, by though we have increased the; that means, level here and here you see. That means, it is almost; that means, this gaps are almost same for two levels, whether there is two level or three levels.

(Refer Slide Time: 26:43)



Delay Optimized CLA: Lee-Oklobdzija '91

(a.) 2-level BCLA Δ=8.5nS (b.) 3-level BCLA Δ=8.9nS

So, then again these are the; that means, another; that means, example of that if I choose this delay optimized CLA circuits. So, at that an I can get the benefit. So, this is another example where we choose something like this 1, 1, then 2, then 1, then 2, then 1, then 2. So, 1, 1 means, this is the one input one input ok. So, this takes two input, then again 2 here is 1 and 1 coming over here, then 1, 1, 1, then 1, 1, 1; that means, this is a 3 input and then 1,1, then this then again 3. So, something and finally, we are having 6 at these particular things. So, that was a maximum fan in is 6 a over here ok.

And for this, for this particular case here you see, this is also the thing is that the maximum level is 6. But here, what we are doing; that means, the somewhere here I i am having 5. In this particular case I am having 5 and here I am having 6, but the levels if I consider, so, the levels if I consider then 1, 2 and then 3. But here, the delay is 8.5. But here, in this particular case the delay is 8.9. In the previous slide also, if you just see it has slight difference. If we increase the level it has slight difference from the 2 level, this CLA optimization.

So, then for today’s class this is it. In the next day, we will see this carry select adder.

Thank you all.