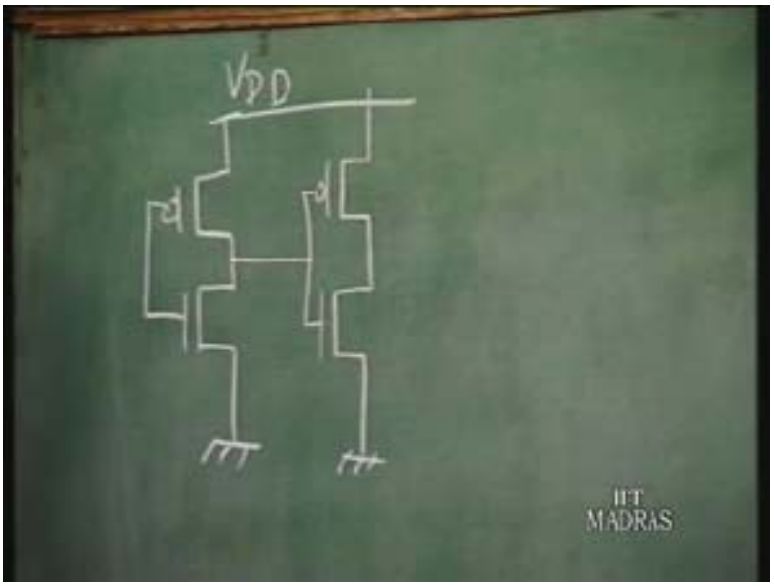
Digital Integrated Circuits
Dr. Amitava Dasgupta
Department of Electrical Engineering
Indian Institute of Technology, Madras
Lecture No # 30

BiCMOS driver; BiCMOS 32-bit adder

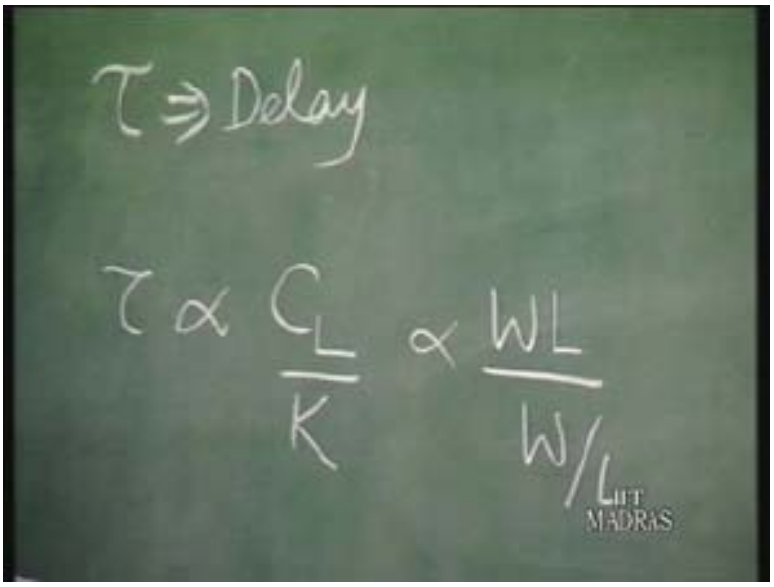
We shall continue our discussion on BiCMOS logic circuits and keep comparing the performance with respect to CMOS and see how actually it is an improvement on CMOS in quite a number of applications. So before we do that let us see how in a CMOS environment, one would drive a large capacitive load? Because we have already seen that a BiCMOS would be an advantage, if we are driving a large capacitive load because the delay does not increase at the same rate as it does in the case of the CMOS. So first if you want to actually make a comparison, we have to see how you use the CMOS to drive a capacitive load.

(Refer Slide Time: 02:11)



This is the CMOS inverter. Before we go into that let us take up a small example. You have a CMOS inverter driving a similar CMOS inverter, that is you have an inverter. Suppose the delay of this first inverter is given by tau.

(Refer Slide Time: 03:44)

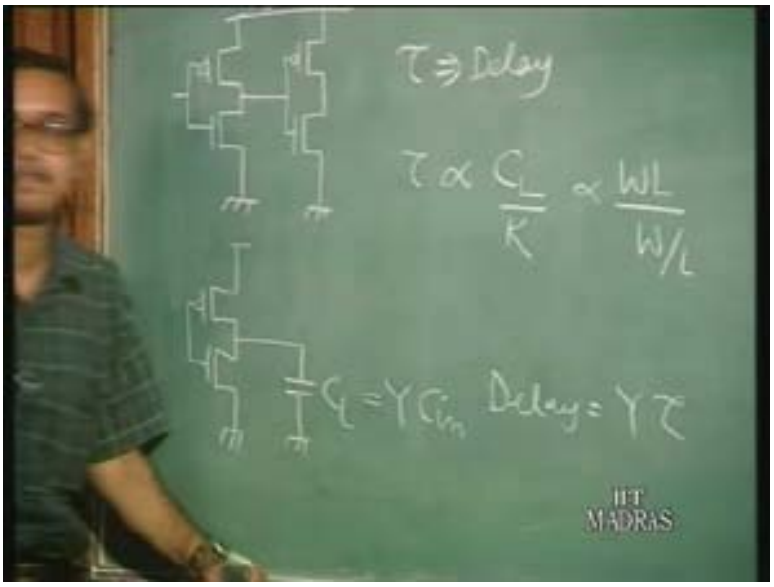


Suppose we increase the widths of all the transistors in this circuit. That is there are 4 transistors by a factor alpha say. That is we make whatever is the width of this transistor, we increase all the 4 transistors. What is going to be the new delay? Is tau going to be changed or is it going to remain the same? That is the question. Now you see that what happens to the capacitance as seen by the first inverter? The capacitance goes up by the factor alpha and you know that the delay, this tau is proportional to, we have seen in all the expressions that it is dependent on the load capacitance by the trans conductance parameter. This C_L is proportional to W into L ratios. This one will be again proportional to W into L and K is proportional to W by L.

What is going to happen is if you increase the widths of all the transistors, C_L refers to the input capacitance of this inverter that is going to go up because W has gone up and the k factor or the trans conductance parameter of the driving transistor, that also goes up by the same factor with the result that the delay remains unchanged. That is again, if you increase the widths of all the transistors by the same amount, the delay doesn't change because the capacitance is going up as well as the driving capacity is going up.

The current charging the capacitance is also going up by the same amount but if you take the lengths of the devices, suppose you increase all the lengths of the transistors by a factor alpha say what is going to happen? The capacitance goes up by a factor alpha, the current goes down by a factor alpha because the current charging the capacitance is proportional to W by L. The delay goes up by the factor alpha squared. If you look at it from another angle, if you reduce the lengths by a factor alpha in fact a speed is going to go up by a factor alpha squared because the delay is going to go down by a factor alpha squared. That is the driving force behind reducing the dimensions. The lengths of the transistors must be made as small as possible from the delay point of view.

(Refer Slide Time: 06:52)



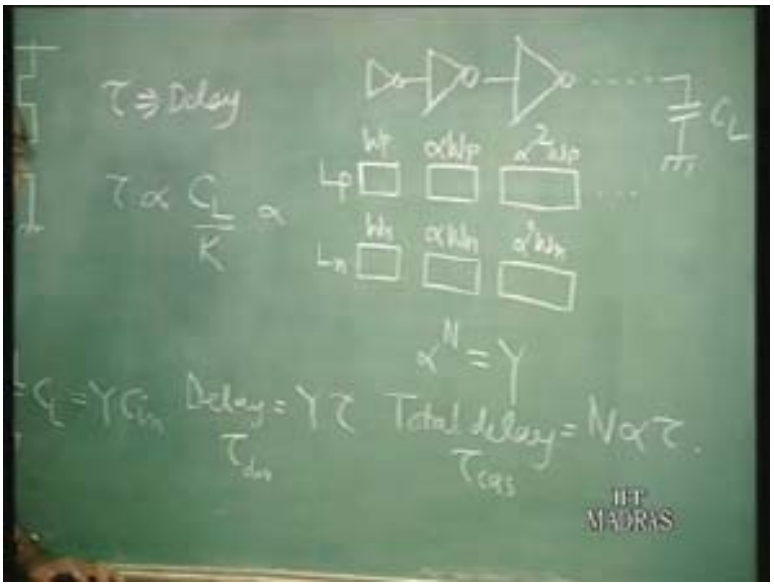
The tau is the delay for this inverter. If it is driving a similar inverter, let us call that delay equal to tau. Suppose this same inverter is driving a capacitive load C_L say and suppose this C_L is equal to Y times the C_{in} of the same inverter, that is the input capacitance of the same inverter. What is going to be the delay? What has happened is that capacitance has gone up by a factor Y. If the delay is tau for driving capacitance equal to C_{in}, if we are driving a capacitance CL, obviously the delay is going to be Y times tau because the delay is proportional to C_L by K, K remains the same, you have same inverter driving a load. In this case the delay is going to be Y times tau. This is a scheme, you have a CMOS inverter driving a capacitive load.

Now let us see how in fact we can reduce this delay. Suppose here itself, instead of having this inverter driving the load directly, we have this particular scheme. That is you have a series of inverters, a cascade of inverters like this. One inverter like this then you have an inverter like this. When you have large number of inverters one after the other driving this loads C_L, this is our original inverter and we make it this way that suppose the original inverter is like this. That is the PMOS and this is the NMOS. The PMOS you have, the length of the PMOS LP and the width is WP and the NMOS here, the length is Ln and the width is W_n. Then the next inverter what we do is we retain a same length but we make this width alpha W_p and we make this width alpha W_n. That is increased by a factor alpha.

What is the delay of this inverter driving this next inverter? That is alpha times tau, because the width of the transistor it is driving or the transistors in the inverter it is driving is increased by a factor alpha. We have already seen that, so this is alpha tau. The next inverter is say alpha squared W_p and alpha squared W_n. What is the delay of

this inverter, the second inverter driving the third inverter? Here what has happened is the widths of the transistors in the third inverter is alpha times the width of the transistors in the second inverter.
Again the delay is going to be alpha times tau. If this had also been alpha times W_n and alpha times W_p, the delay would have been tau itself. We have seen that, if you have equal widths, it is the same tau. If you have n such stages so that you have n such stages and alpha to the power N is equal to Y where Y is C_L by C_{in}. This is alpha times the capacitance of this input capacitance. This input capacitance of the second stage is alpha times this. The third one is alpha squared times C_{in} and the final C_L is going to be alpha to the power N times C_{in}. That is the capacitance.

(Refer Slide Time: 11:50)



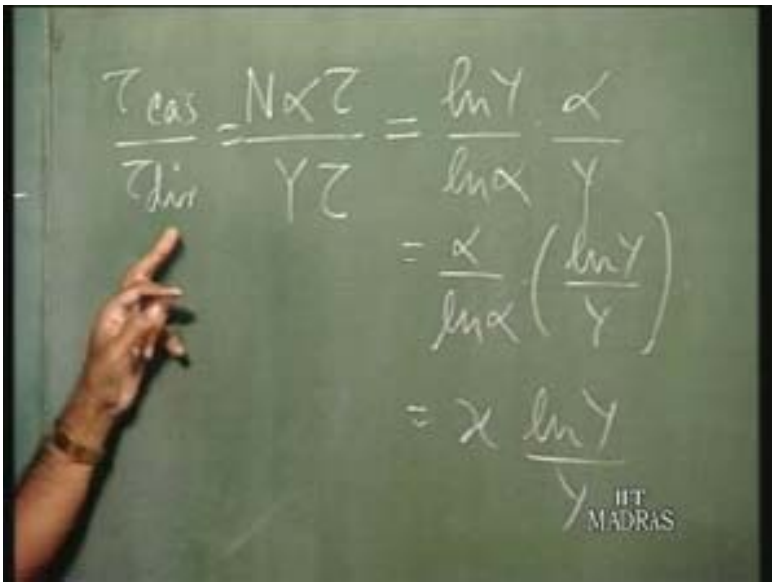
You have alpha to the power N, if you have N stages. You have alpha to the power N is equal to Y where Y is the ratio of C_L to C_{in}. What is a total delay in this case? You have n stages and each stage has a delay of alpha times tau, so the total delay is n times alpha tau. In this scheme total delay is equal to N times alpha tau. You see that in this, if you have cascade of inverters like this, the total delay is N times alpha tau whereas if you are driving it directly it would have been Y times tau.

Now let us see. Basically we will call this cascade. Total delay will call it tau cascade, cascade of inverters and tau direct. Y tau is the delay of the direct, when it is directly driven. Tau cascade by tau direct is equal to N times alpha tau divided by Y times tau. What is N actually? N we know is alpha to the power N is equal to Y. We know that alpha to the power N is equal to Y, so we can write N ln alpha, if you take the ln on both sides is equal to ln Y. N is equal to ln Y by ln alpha. Here we have tau cascade by tau direct becomes equal to ln Y by ln alpha into alpha by Y. That is equal to alpha by ln alpha into ln Y by Y. We call this say x times ln Y by Y. See this ln Y by Y is a

constant depending on the capacitance you are driving. If Y is the ratio of the load capacitance to the input capacitance, which is a constant.

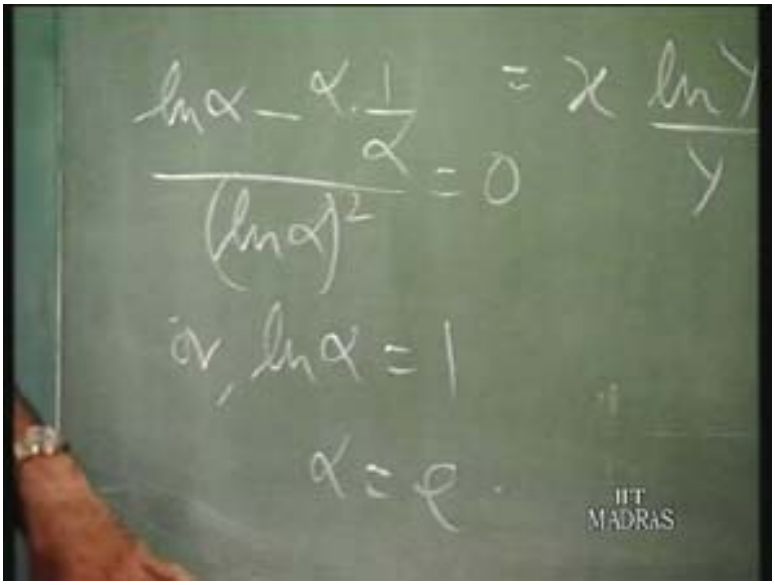
Now what you have to do in our design? We have to choose the number of inverters we want to put and the ratio of the widths of the inverters that is the design which we have to take up.

(Refer Slide Time: 14:01)



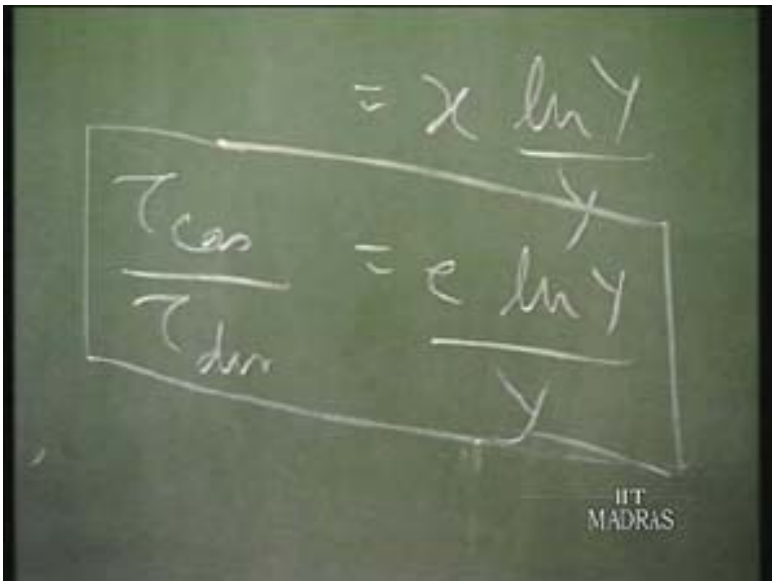
What is going to be the factor alpha? That is what we have to see. Obviously what our aim should be to reduce this as far as possible. Tau cascade by tau direct should be as small as possible. We have to minimize the value of x and what is x? Alpha by ln alpha, so alpha by ln alpha we have to minimize. We have to differentiate this alpha by ln alpha and put it to zero. If you differentiate this alpha by ln alpha, what do you get? You have ln alpha minus alpha and which means that ln alpha is equal to one which means that alpha is equal to e.

(Refer Slide Time: 14:51)



If this is the case, if alpha is equal to e then what is this tau cascade by tau direct? Alpha is equal to e, so alpha by ln alpha is equal to e. alpha is equal to e, ln alpha is equal to one so that is going to be e.

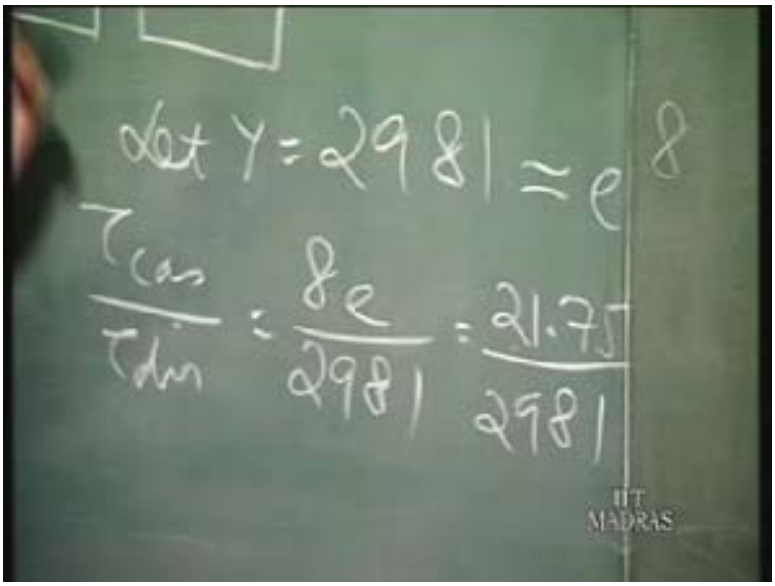
(Refer Slide Time: 15:40)



Basically it becomes e times ln Y by Y. That is the tau cascade by tau direct. We can take an example and see how much it is effective. Say suppose just we are taking some numbers.

Let Y is equal to, I am just using an arbitrary number. This 2981 is taken because that is equal e to the power 8. ln Y is going to be equal to 8, so tau cascade by tau direct is going to be 8 e by 2981 and 8 e is around 21.75 by 2981.

(Refer Slide Time: 16:46)



What you see is, if you are using a cascade of inverters, you have substantial savings in delay. This is less than 1% of the delay you would normally have had, if you would have been driving the load directly. If you had been driving the load directly using that CMOS inverter, whatever delay you would have had by having a cascade of inverters like this, the delay is going to be about less than 1% of that delay. Usually in a CMOS environment, when you are driving large loads, you have to use cascade of inverters like this. Especially this type of circuit is important.

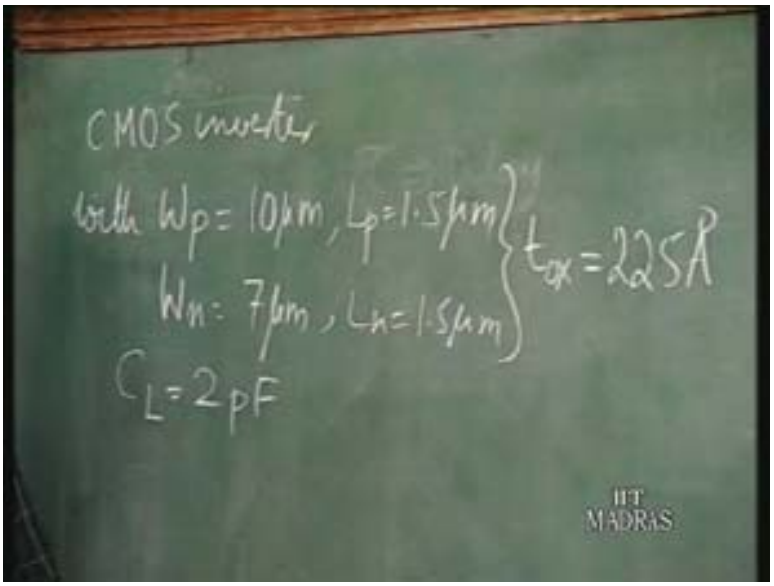
For example in a clock generator circuit, you know that in a large circuit you will have a master clock somewhere and that clock output may be going to a large different regions in the circuit. It may be driving different parts of the circuit. The effective load for that clock generator is going to be enormous. You will have to have such drivers for that because it is important because if the delay is too much, basically you have to use a slower clock. You cannot go for very high frequency clock. In order to reduce the delays, you have to go for buffering or use the cascade of inverters like this with gradually increasing dimensions. The inverters will have gradually increasing dimensions.

In this particular case you will have 8 such inverters followed by the load because you have to reduce the delays. On the one hand, you are reducing delay but of course you are paying for it in terms of area. That is you require so many inverters not only that the sizes of the inverters are gradually increasing so that the last inverter will be a very big inverter, the widths of the transistor is going to be very large because it is increasing in a geometric fashion.

What we shall now do is we shall see how the BiCMOS can be effective actually in driving such large loads by saving area when compared to the CMOS counterpart. We shall take an example for that and try to understand with that. Suppose you have an CMOS inverter. The different dimensions are W_p is 10 micron, W_n is 7 micron and of course L_P is 1.5 micron and L_n is 1.5 micron also. This is of course as you see that, the lengths of the devices are always kept at the minimum value. We only play around with the widths because as we have seen, the delays are mostly determined by the lengths.

For this inverter, suppose it has to drive a load capacitance of 2 picco farad and also let us say that this in this CMOS technology, the oxide thickness is given as 225 Armstrong. That is the specified, 225 Armstrong is the oxide thickness. These are the dimensions of the devices, this is the load capacitances which it has to drive.

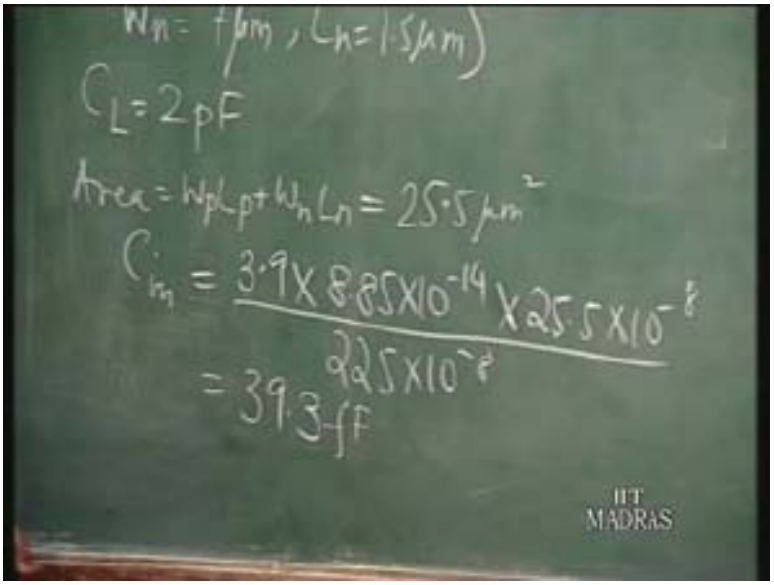
(Refer Slide Time: 21:10)



We shall see in our design, how many stages of CMOS we require for driving this load and what are the sizes or the dimensions of the different transistors? First what we have to calculate is the C_L to C_{in} ratio that is the input capacitance of this particular inverter. We have to take the total area. What is the total input area that is W_p into L_p plus W_n into L_n. Area is equal to W_p into L_p plus W_n into L_n is equal to, so 10 into 1.5 is 15 and 7 into 1.5 is 10.5, so it is 25.5 micron squared.

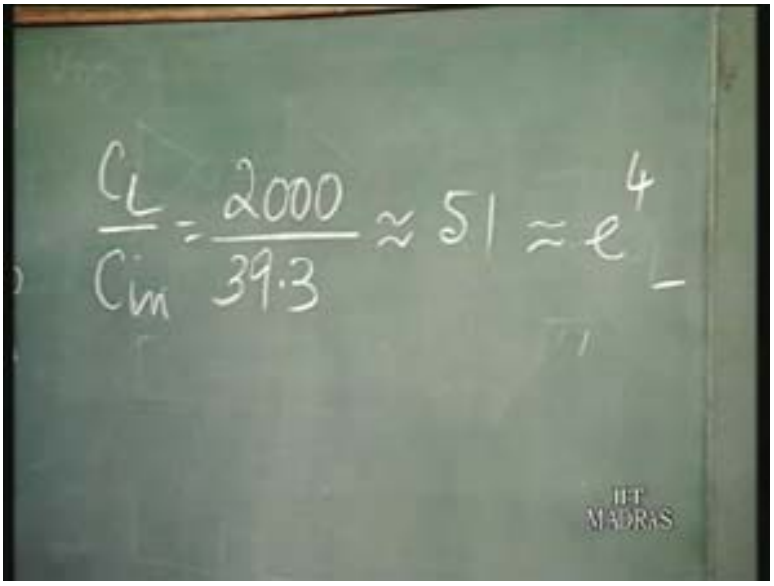
C_{in} is equal to epsilon a by TOX. What is epsilon? For silicon dioxide it is 3.9 into 8.85 into 10 to the power minus 14. That is epsilon$_0$, this is epsilon R into area 25.5 into 10 to the power minus 8 per centimeter. This is in centimeters, all in CGS units, this farad per centimeter, this is centimeter squared by oxide thickness 225 Armstrong's, 225 into 10 to the power minus 8. This comes to 39, I just give the result 39.3 femto farad. This is 39.3 femtofarad so the load capacitance is given as 2 Pico farad.

(Refer Slide Time: 22:51)



C_L by C_{in} is equal to 2000 by 39.3. This comes to almost 51 which also happen to be almost equal to e to the power 4.

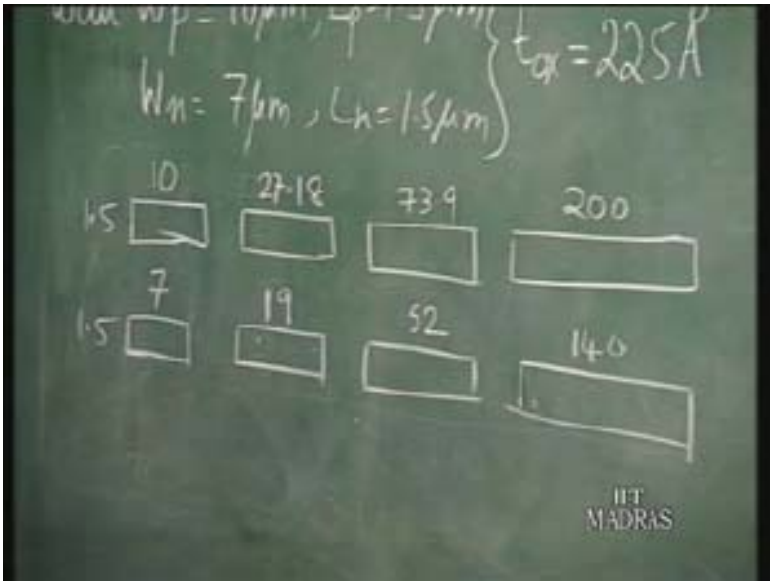
(Refer Slide Time: 23:35)



What is going to be the dimensions of the individual transistors? This is W_p, this is W_n, so this is all 1.5. The lengths do not change, this is all in microns, W_p is 10 slightly wider than W_n in this case, seven here W_n.

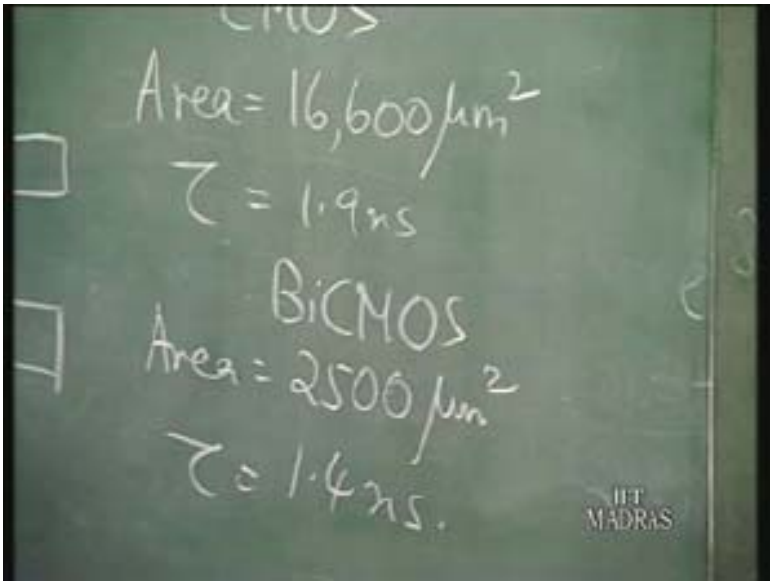
The next device is going to be e times. All this what we have done is we have taken that the widths of the transistors is going to be e times increasing by a factor e. This is going to be wider, so this is going to be 27.18 and this is going to be 19. Then the third is going to be almost 73.9 and this is going to be 52. Then the next one this 200 and this is going to be around 140. For all the transistors, the lengths remain same as 1.5 micron. These are the 4 stages. From here to here it is going to be e, e squared, e cube and the load is again another e that is e to the power 4. C_L by C_{in} is equal to e to the power 4. So you have this 4 stages driving the capacitances.

(Refer Slide Time: 25:55)



Just I am quoting from a report, from which I took this example. They were driving the same load using CMOS. They had to fabricate this, they required an area of 16600 micron squared and the delay was 1.9 Nano seconds. They fabricated the same thing using a single BiCMOS inverter where the MOS dimensions where the same as this first inverter and they just included a bi polar transistor just as we had discussed last class that you have the CMOS with a bi polar transistor, normal bi CMOS inverter where the MOS dimensions are given by the smallest device here. This is for CMOS. For a BiCMOS, the area required was just 2500 micron squared and the delay was 1.4 Nano seconds. See you are driving a load which is 51 times the input capacitance.

(Refer Slide Time: 27:21)



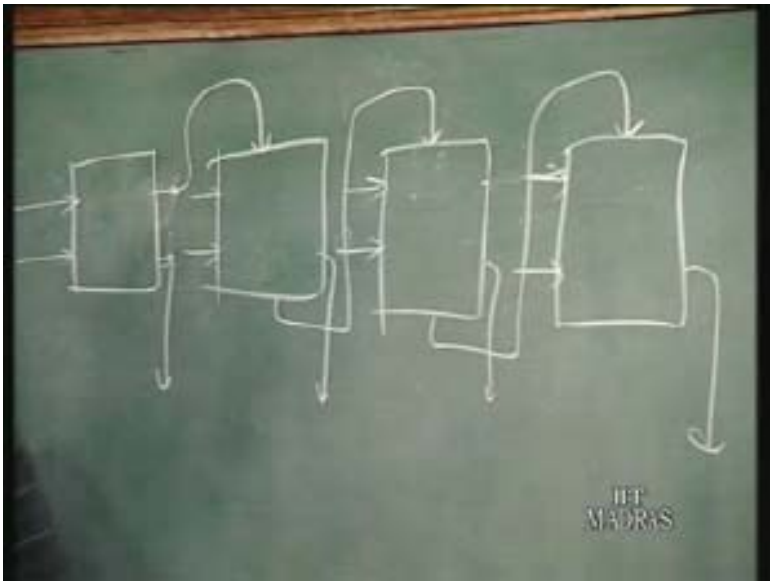
If the bi polar transistor is able to give you a current gain of more than 50, when you compare with the same CMOS inverter driving the same load, you are going to get a lower delay. Isn't it? You see that this in fact gives you a lower delay compared to the CMOS inverter. Not only that, the area requirement is much smaller because you are using just the smallest size devices here and you just have extra some bi polar transistors whereas here you use progressively larger devices which requires a lot of area. So using BiCMOS drivers to drive large capacitance loads, you not only gain in terms of speed but also you require a much smaller area. That is the advantage of BiCMOS drivers over CMOS drivers and it is now being widely used wherever you want to drive.

For example in memories, we shall see that also. At the address decoder you have the memory cells to be driven. If you have a one mega bit memory, you will have 10 to the power 6 cells which may be 1000 by 1000 array. The output of an address decoder must drive thousand gates which mean a large capacitive load and BiCMOS driver is obviously going to be much superior compared to the CMOS driver. There are many applications where you require large capacitive loads to be driven and obviously the BiCMOS inverter or BiCMOS driver is going to be superior in that respect. That is the example of a BiCMOS driver. We shall take up some more examples where BiCMOS is used. Here what we have taken is the conventional BiCMOS inverter as a driver.

Instead of the CMOS inverter you use a BiCMOS inverter or you may use instead of the CMOS gate for example a NAND gate, you may use a BiCMOS NAND gate where the output you have a transistor which with the help of the current gain, it gives you lower delays but in many other applications it is the judicious use, you do not use the BiCMOS as such as gates but by the judicious use of bi polar transistors, you can increase the speed of the operation. We shall take up one more such case. In the case of an adder circuit and we shall see how to incorporate a bi polar transistor in an essentially CMOS environment. You get higher speed.

The next example we shall take up is an adder circuit. I shall just give you a brief introduction about adder, just as to remind you the carry look ahead adder. I think you are aware of it but anyway just to give you an idea what is a carry look ahead adder.

(Refer Slide Time: 32:19)



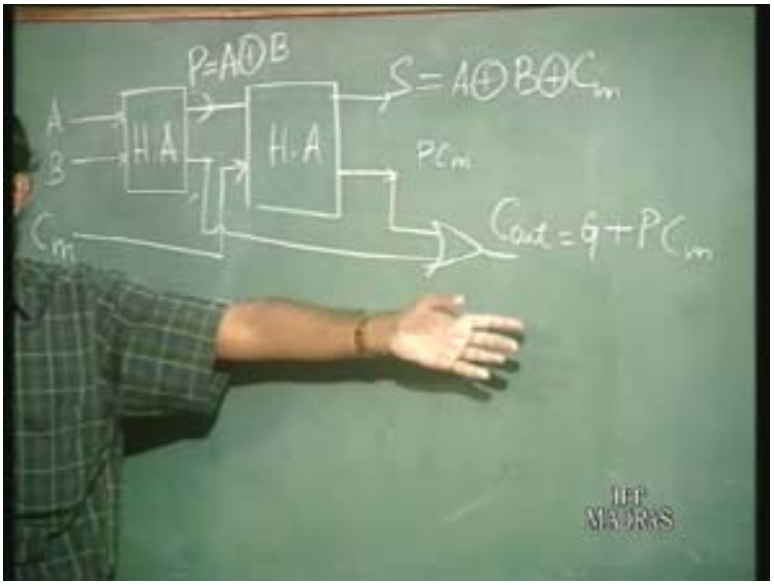
Normally you have two types of adders, one is called the ripple carry adder where if say for doing a four bit addition, you have four adders and what you do is you have the inputs to each stage and the output of each adder is going to be the sum and carry outputs. What is going to happen? You have the some output say coming out and the carry output is going to the next stage. It is just rippling through all the stages. If you have a 32 bit adder, then the output of this has to go here then the next will change because of that and so it is a rippling effect and it takes a long time. What the carry look ahead adder does is, it prepares this carry inputs to each stage before and in a very short time so that even for the last bit, the addition does not take much time.

We shall just see what a carry look ahead adder is and then we shall see how incorporation of bi polar transistors in a CMOS environment actually can speed up the operation. Basically a full adder which has got three inputs that is the two inputs to be added and a carry input which comes from the previous stage. A full adder can be constructed using two half adders. What is a half adder? The half adder has got two inputs and two outputs. One is the sum output and the other is the carry output. If the inputs are 0 0, sum is 0, carry is 0. If one input is 1, the other is 0, sum is 1, carry is 0. If both the inputs are 1, sum is going to be 0, carry is going to be 1. Basically the sum output which you have is the Exclusive OR function of A and B which you consider the inputs as A and B and the carry output is equal to AND function of A and B.

Now let me call this Exclusive OR of A and B, we call it the P term equal to A Exclusive OR B. Why we call it P, I will just come to that and this is called G equal to A B. So P stands for propagate and G stands for generate. You have the propagate term and G is the generate term. Now you feed this directly to the next adder. The carry in which you have to this stage goes in as the other input to this half adder. Here the output you have, sum is equal to Excusive OR so this half adder again, so the sum output here is going to be the Exclusive OR of the two inputs here. This is Exclusive OR of A Exclusive or of B Exclusive or of the carry input and here what you feed is (Refer Slide Time: 35:05), so here again this is going to be the AND function of the two inputs. Here one of the input is P and the other input is C_{in}. P C_{in} and you take an OR of this two carry outputs. The carry output is actually equal to G plus P into C_{in} where G is equal to AND of A and B and P is equal to Exclusive OR of A and B.

Now G is called generate because it generates a carry. It is going to be high if both A and B is going to be high. It generates a carry. If A and B that is both inputs are high, of course you are going to have a carry, irrespective of carry in or whatever. You are going to have a carry out, so it generates a carry. You are going to have a carry out if G is high. The other case where you can have a carry is that you had a carry from the previous bit that is there was a carry input and you are propagating that carry. That is there was a carry input from the previous bit and now if one of the inputs, other inputs A or B is high then you basically propagate that carry to the next bit. P is going to be high it's an Exclusive Or function, so if A or B is high, any of them is high, P is going to be high. If C_{in} is high and either of A and B is high, you have the carry output. Basically the carry input is propagated to the output. P is going to be high if A Exclusive Or B is high. You have a carry output if either you generate a carry in this present bit or you propagate whatever carry in, carry in is just propagated to the output. If P is high and C_{in} is high then also you going to have a carry output. This is for a particular bit, you have the carry output.

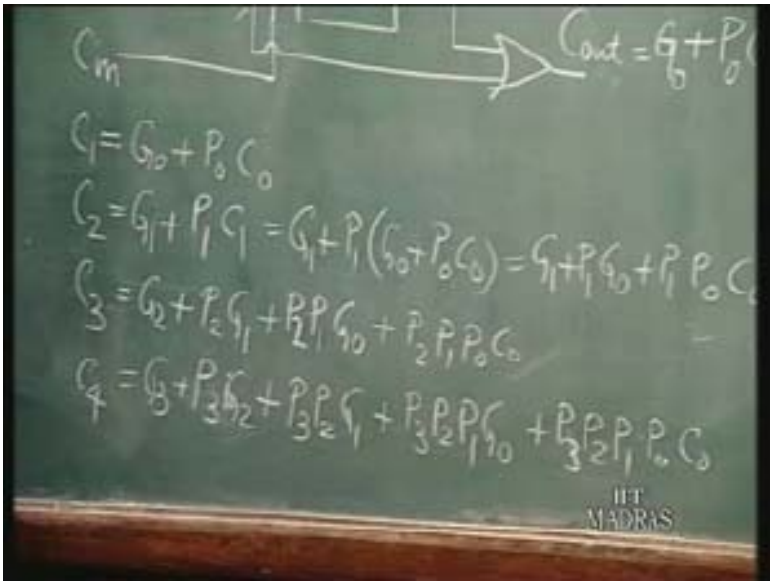
(Refer Slide Time: 36:20)



Now if you consider say a four bit adder what you can do is you can think of it this way. The input you have C_0 is the carry input to the four bit adder. For the first bit you have A_0 and B_0 at the inputs and basically you have which gives rise to G_0 and P_0. Basically if you have this all this zeros here, you have P_0 and you will have G_0, everything is zeros. If it is for bit one this one will be A_1 and B_1, you have G_1 and P_1 where G_1 is AND function of A_1 and B_1 and P_1 is Exclusive OR of A_1 and B_1. What is C_1? C_1 is G_0 plus P_0 C_{in} that is actually the C_{in} or I think we should put it C_0 which is actually C_{in}. This is C_1 which goes as the input to the first bit, C_1 is G_0 plus P_0 C_0. For first bit you have the inputs A_1 B_1 and C_1, bit one. This is for bit zero.

For the next C_2 will be equal to G_1 plus P_1 C_1 and what is C_1? C_1 is G_0 plus P_0 C_0. We can write G_1 plus P_1 into G plus P_0 C_0 which means this is equal to G_1 plus P_1 G_0 plus P_1 P_0 C_0. You can see the logic here that C that is going to be a carry output or the carry input to the bit two, if there is a carry generated at bit one or there was a carry generated at bit zero which got propagated through P_1 or there was an input carry which got propagated through both P_0 and P_1. That is how you get C_2. Similarly we will get C_3, you can do it the same way. It will be G_2 plus P_2 C_2 and then you instead of C_2 you write the same thing.

(Refer Slide Time: 41:51)

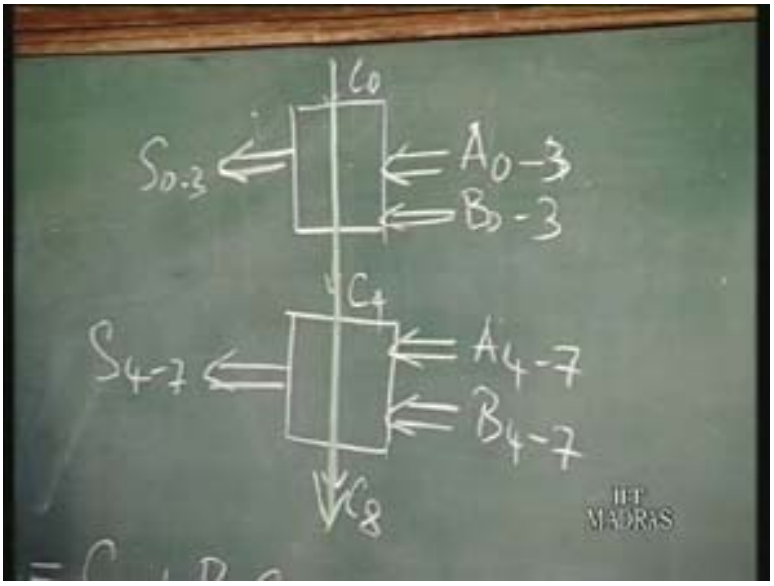


What you will get? You will get $G_{2.}$ plus $P_{2.}$ $G_{1.}$ plus $P_{2.}$ $P_{1.}$ $G_{0.}$ plus $P_{2.}$ $P_{1.}$ $P_{0.}$ $C_{0.}$. Similarly $C_{4.}$ you will get as $G_{3.}$, I am just writing it down, plus $P_{3.}$ $G_{2.}$ plus $P_{3.}$ $P_{2.}$ $G_{1.}$ plus $P_{3.}$ $P_{2.}$ $P_{1.}$ $G_{0.}$ plus $P_{3.}$ P_{2} $P_{1.}$ $P_{0.}$ $C_{0.}$. So these are the terms. The interesting thing here is that what you can do? You already know these things. Instead of $C_{4.}$, you see if it was a ripple carry you have to go through all the stages before you do it but here you can generate these things.

Basically this scheme which you are going to have now is suppose we have a 32 bit adder. What is done is see if you have a 32 bit adder, this is going to become a very cumbersome expression to do it. So you break it up into say 8 4 bit adders, carry look ahead adders. You have 1 4 bit carry look ahead adder, so here you have the inputs say $A_{0\text{-}3.}$ and $B_{0\text{-}3.}$ and you have the output $S_{0\text{-}3.}$. So $C_{0.}$ comes in here and $C_{4.}$ goes out and then this is another carry look ahead adder, where you have the inputs $A_{4\text{-}7.}$, $B_{4\text{-}7.}$ and you have the outputs $S_{4\text{-}7}$ and $C_{8.}$ goes out.

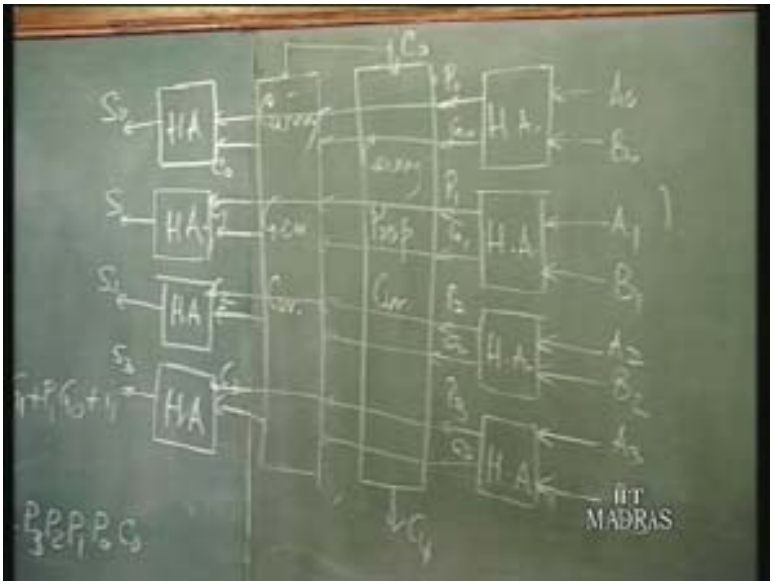
Basically this is the four bit carry look ahead adder and then you are going to have eight such carry look ahead adders. What is a critical path in this circuit? The critical path is this transmission of this carry, it has to go from $C_{0.}$ so to the eighth adder or this eighth 4 bit carry look ahead adder. So only when this is available made to that carry look ahead adder, that output can be a valid one.

(Refer Slide Time: 43:52)



This is the critical path for this carry look ahead adder and this must be properly designed because the total delay of the 4 bit carry look ahead adder is going to depend on this path. How fast this is? Basically if you look at it this way, the 4 bit carry look ahead adder will consist of, I just draw it like this. You have four half adders. You have $A_{0.}$ $B_{0.}$ fed here, $A_{1.}$ $B_{1.}$ fed here, $A_{2.}$ $B_{2.}$ fed here, $A_{3.}$ and $B_{3.}$ fed here and what is coming out of this half adders? You have the P's and G's, you have $P_{0.}$ and $G_{0.}$ here, you will have $P_{1.}$ and $G_{1.}$ $P_{2.}$ and $G_{2.}$ and $P_{3.}$ and $G_{3.}$.

(Refer Slide Time: 49:47)



So all this is going as input to a carry propagation circuit. This is the carry propagation circuit which has its input C_0 and passes out C_4 which is going to be used by the next 4 bit carry look ahead adder. As we have seen that in order to evaluate C_4 what you require as inputs, if you look at this expression here what you require as inputs is C_0 as well as all the G's and P's. If this is available you can generate C_4. That is what it requires at the input then you have another circuit which is called the carry generation circuit which basically generates the carries for the different bits in this 4 bit carry look ahead adder.

You are going to generate the carry that is C_1, C_2, C_3 as well as of course you require C_0 also. What does it require as an input? It requires input as C_0, again C_0 is required and the different P's and G's up to that state. So again you have the different P's and G's coming to this circuit also and you have C_0 also coming here and what you get out of this circuit is, you will have the different carries C_0, C_1, C_2 and C_3 and this when you have another bank of four half adders, this is another set of four half adders which you have and this half adders require P's and the C's as a input. We have seen the model for full adder where sum is equal to Exclusive OR of P and C where P is again the Exclusive OR of A and B that is already available. This is going to give you s_0, again for this half adder you have the inputs P_1 and C_1 is required.

S is equal to Exclusive OR of P_1 and C_1. Basically you just require the Excusive OR here because the carry in is here. You have S_0, S_1, S_2 so this is again the P's are extended up to this. So you have S_1, S_2, S_3. This is a four bit carry look ahead adder where you have the input C_0, you getting all the input bits A_0 to A_3 and B_0 to B_3 and what you are getting at the output is S_0, S_1, S_2, S_3 and the output carry C_4. Now as we said, the crucial part of the circuit which gives rise to the delay is the carry propagation circuit because what it is doing is, it is taking the carry in and it is passing out the carry in.

Here the carry generation circuit is actually a subset of this particular circuit you can say because if you look at the expressions $C_{1.}$, $C_{2.}$, $C_{3.}$ it is the simpler expression compared to $C_{4.}$ and you are just using the same things, all the P's and G's and the $C_{0.}$'s. So in this part of the circuit you have two half adders and then you have the two carry propagation circuit and the carry generation circuit.

The delay's in the carry generation circuit that is $C_{0.}$ is already there, $C_{1.}$, $C_{2.}$, $C_{3.}$ that is obviously going to take less time than $C_{4.}$ and $C_{4.}$ is very crucial because as we have seen that in a 32 bit adder, you are going to have 8 such four bit carry look ahead adders and the delay, the $C_{4.}$ has to go in as a next four bit adder and as the input of the carry propagation circuit of the next four bit adder and you will generate $C_{8.}$ which has to be passed on to the next four adder to generate $C_{12.}$ and so on. This is the most crucial part in say a 32 bit order. How fast you can generate the carry in this carry propagation circuit?

We shall see in the next class how this is generated in a CMOS environment and how you can improve the circuit or basically reduce the delay by using bi polar transistors to create a BiCMOS type carry propagation circuit and we shall see that by just using very few bi polar transistors, it is not that you are replacing a CMOS NAND gate by a BiCMOS NAND gate or a CMOS inverter by a BiCMOS inverter but by the judicious introduction of bi polar transistors you can improve the delays. It is you have to see where you have you want to introduce that bi polar transistors, how it can improve the speed. We shall take that up in the next class.