CMOS Digital VLSI Design

Prof. Sudeb Dasgupta
Department of Electronics and Communication Engineering
Indian Institute of Technology – Roorkee

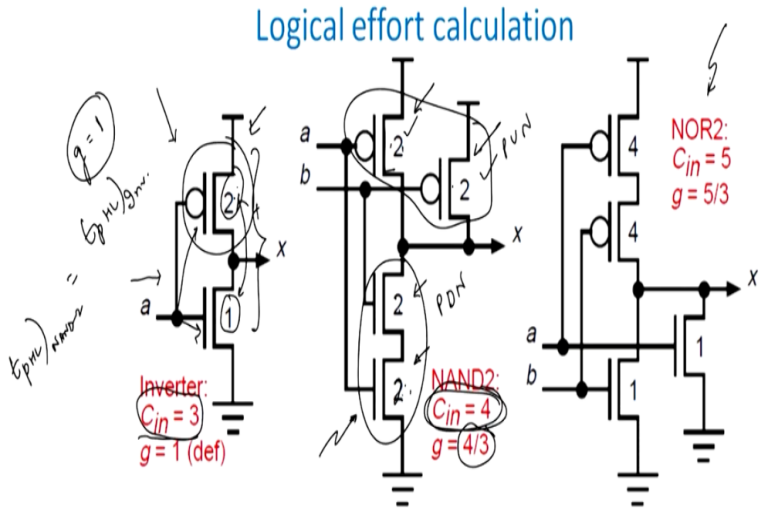
Module No # 05
Lecture No # 23
Logical Efforts – II

Hello again welcome to the NPTEL online certification course on CMOS digital VLSI design and we start the second module of logical effort we have finished the first module and we saw what is the meaning of logical effort in the first module. We also understood why logical effort is important in calculating the delay between the primary input and output of a system what will be doing in this module is primarily looking into how to calculate logical effort and then using logical effort how can we calculate the total delay right.

So the we had discussed what is logical effort in the previous module and we will be discussing how to calculate or evaluate the logical effort in this module. Now let us look at how to calculate first of all the logical effort for any of the conventional gates available to you may be it is 2 inputs, 3 input, n input, NAND gate, NOR gate, XOR gate whatever gate combinational logical let us see how to do that.

So as a discussed with you logical effort is primarily the ability of a particular gate to drive the same amount of output current as a minimum size inverter right. So I have inverter whose low to high and high to low as been made equal by making it my making it 2 is to 1.

(Refer Slide Time: 01:56)



Logical Effort: Designing Fast CMOS Circuits Ivan E. Sutherland Bob F. Sproull David L. Harris

So if you look at this slide here in this slide you will see that this is basically made 2 is to 1 is primarily means that so therefore is you look from input side right if you look from input side it is C in which is the input capacitance is basically 2 + 13 this is 2 +1 3 because any input has to drive this PMOS as well as it has to drive this NMOS right and therefore the input capacitance is basically 2 + 1 3.

So I have my C in to b equals to 2 + 1 3 so what we get from here is that we define therefore the logical effort of single inverter g to be equals to that so if it is 2 by 3 and if you have a single inverter available with you then we define the g of this is to be equals to 1 which means that with respect to self of with respect to itself it will drive exactly the same amount of current in the output load provided 2 is to 1 ratio is maintained between the PMOS and the NMOS.

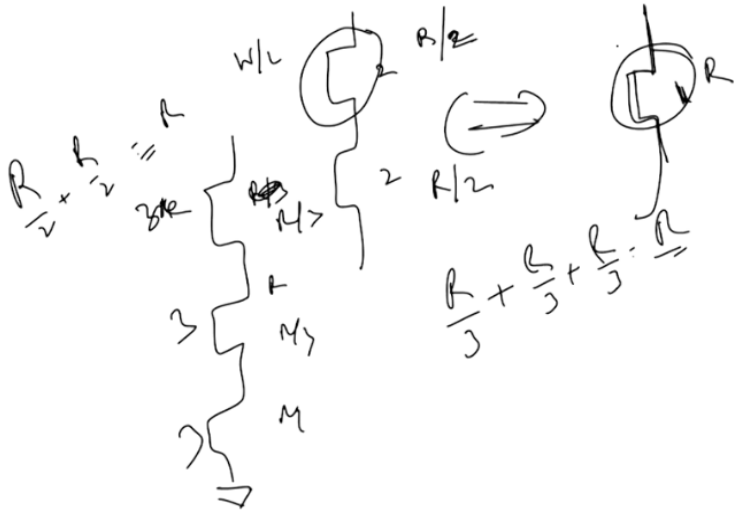
So any inverter with the sizing of 2 is to 1 between PMOS and NMOS will have its logical effort g = 1 that is the basic definition of the logical effort. So calculation of logical effort for a basic CMOS right is straight forward and simple and value is equals to 1 provided the PMOS to NMOS ratio will 2 is to 1. Let us have a look at maybe a 2 input NAND gate which is available here now as you can see here right.

If you look here then NAND has got 2 PMOS's and parallel in the pull up right which is this this and this thing parallel pull up PMOS's and we have got 2 series NMOS's in the pull down so this is my PDN this is my PDN and this is my PUN right PUN. Now why I am why I am making this PUN 2 and 2 aspect this 2 2 and 2 are basically the aspect ratio. I want to make it such that now

the delay high to low delay for a 2 input NAND gate is exactly equals to high to low delay for a simple inverter which is there in this side.

So the inverter so therefore tp high to low of NAND to logic right is exactly equals to tpHL of an inverter a skewed inverter which is shown here why because when you have a W/L ratio 2 and you are in series to each other this effectively come out to a fact that you have only one device with aspect ratio 1 right I will explain to you what do I mean by that which is basically if you look carefully here and I will just show to you what do I mean by that.

(Refer Slide Time: 04:54)



So if I got let us suppose two transistors which are 2 each and they are in series to each other than this is equivalent to saying that I have a single transistor of aspect ratio one I hope you understand this point why is it why is it like that because you see this was W/L also you have double the ratio here so the resistance offered by this transistor is R /2 so suppose this is r when this is as a R/2 this also R/2. So R/2 + R/2 in series will give you r so this is what I what I wanted to tell you that if there.

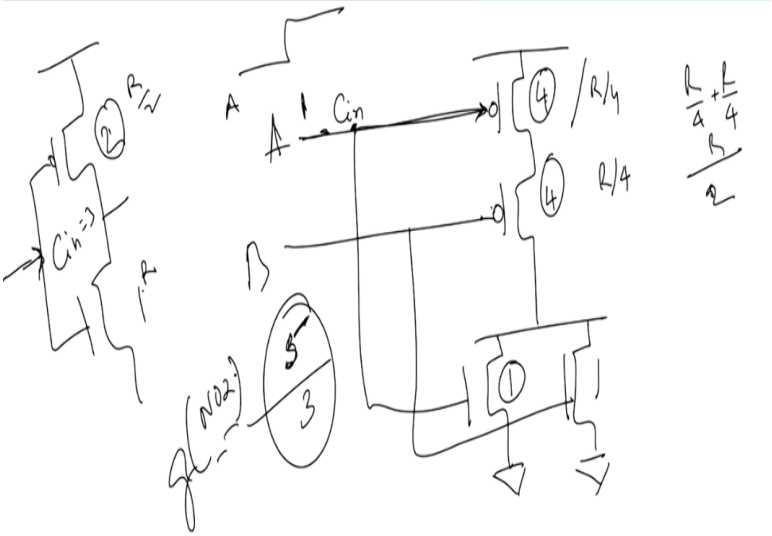
For example if you have a therefore the 3 input NAND gate then you will have pull down 3 in number when this each one will be R/3 so if you make it 3R 3 and 3 aspect ratio and this I will be R/3 and if you add all these R/3 is together we effectively get = R which is nothing but transistor of pull down network of a minimum size inverter right. So this is how we calculate the value of your logical effort or the pull down case right.

So therefore your aspect ratio is 2 and 2 here then let us go to the pull up network is relatively easy because please understand pull up network for low to high propagation either as the 2 has to be switched on not all both of them in does not have to switch on right either of the two PMOS's has to switch on. So when they switch on their W/L ratio equal to the PMOS W/L ratio this one for low to high transition exactly that of the inverter.

So therefore input which is being fed here will actually see this PMOS at it is actually see as NMOS so this is 2 + 2 therefore your input capacitances is basically = 4 right so your input capacitance = 4. And therefore we define the G logical effort to be equals to 4 which is this one this plus this divided by 3 because if a inverter it is in 3. So we define g to be equals to 4/3 I hope you understood how we are calculating the value of the logical effort here.

And the same concept let me again explain to you by the concept of the two logic so I have a NOR 2 logic is basically 2 input NOR gate. In a 2 input NOR gate let me give you an idea in a 2 input NOR gate what i will get is something like that.

(Refer Slide Time: 07:34)



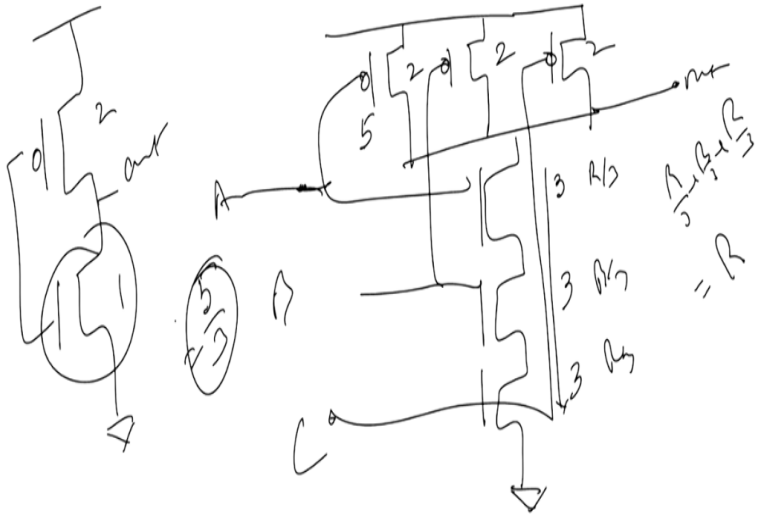
2 input NOR gate will look something like this right and you will have this and then you will have a parallel this right so your are driving this and you have input here and then you have A input and then if this is B then I will be driving here FB here. So I have got 2 inputs A and B now I want to make it exactly = or inverter 2 is to 1 which I have discussed you just now right. So If

you look if you have a signal being fed here it sees a C in to be = 3 let us see what happens in this case I want to make it equal.

So as my discussion was there for a previous case I can also tell you that this will be therefore equals to 4 and 4 why 4 and 4 because if you make it 4 and 4 see this was equals to 2 is this is R this will be R /2 because it will double the width so I want to make it R/2 so each it becomes R/4 then R/4 then if I add R/4 + R/4 I get R/2 as the effective value. So therefore this will be 4 /4 whereas this can offered to the equals to be 1.

I suppose you have understood what I am trying to tell you is that why this is 4 why this is 4 will up to 4 becomes R/4 and therefore if R/4 if you add up we will get exactly the same resistance as offered by the pull up NMOS transistor in the (()) (09:05). So therefore Cin in this case is nothing but 4 so if nothing but 4 so if am inputting input 0 to 1 transition in A right then its input C is a loading of 4 here and the loading of 1 here. So you have 5 is the loading which you see so 5 divided by that 2 + 1 3 so 5 / 3 is the logical effort for a NOR 2 logic is it okay so this is how you calculate the NOR 2 logic.

(Refer Slide Time: 09:36)



So let we explain to you again maybe a NAND 3 logic maybe a NAND3 logic will have PMOS's 3 NAND gate in the up one right this this and then you will have 3 PMOS's in series NMOS is in series and it will be connected to a external loading so I will have A I will have B I will have C. So If you want to make it exactly = sorry if you want to make it exactly = to a the single sized inverter which is something like this right 2 is to 1 ratio and then you have to find out as I

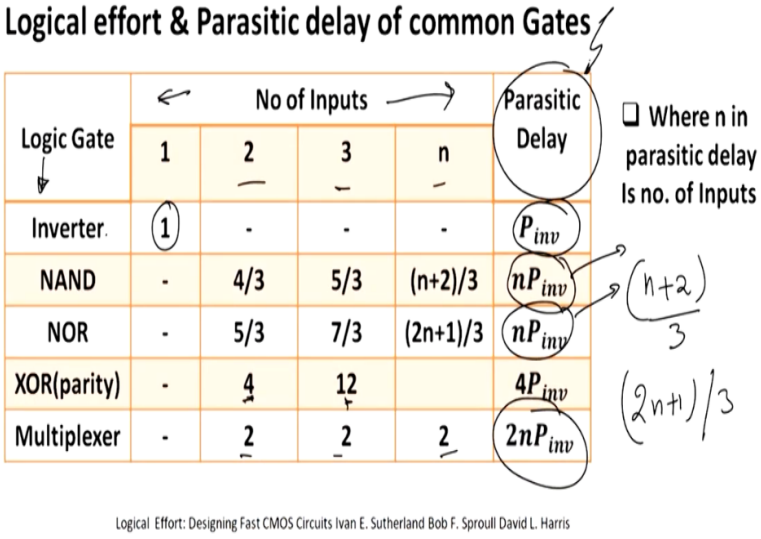
discussed with you this will be tau 2 as to be tau 2 right because then you require only one of the PMOS's switch on for low to high transition.

So this is my output here this is the output here so I have only one of the PMOS's is to switch on and all the PMOS is switch on in order to issue that the low to high transition in the output whereas I want to all my NMOS to switch on to have high to low transition and please understand each one therefore should have thrice the value available to it then this will be 3 and R/3 and this will be R/3.

So R/3 + R/3 + R.3 will give R which is nothing but the resistor offered by the NMOS so a signal fed here will actually see 5 as the input and therefore during 5/3 again the logical efforts for the 3 input NAND gate. So therefore you see more complicated the gate is or larger the number of input profile available to you larger will be its logical effort right it was it is more difficult for the whole gates structure to give the same amount of current as output current as your small seized inverter right.

So this is what I wanted to just this is how you calculate the logical effort for all the other gates right so I get 5 / 3 I get 4/3 for a 2 input NAND I get 5 / 3 for a 3 input NAND and I also get 5/3 for 2 input NOR right. So these are the few things which you should be aware of and this is what I want was trying to say that therefore for inverter it is 1 and if you this is a logic gate.

(Refer Slide Time: 12:02)

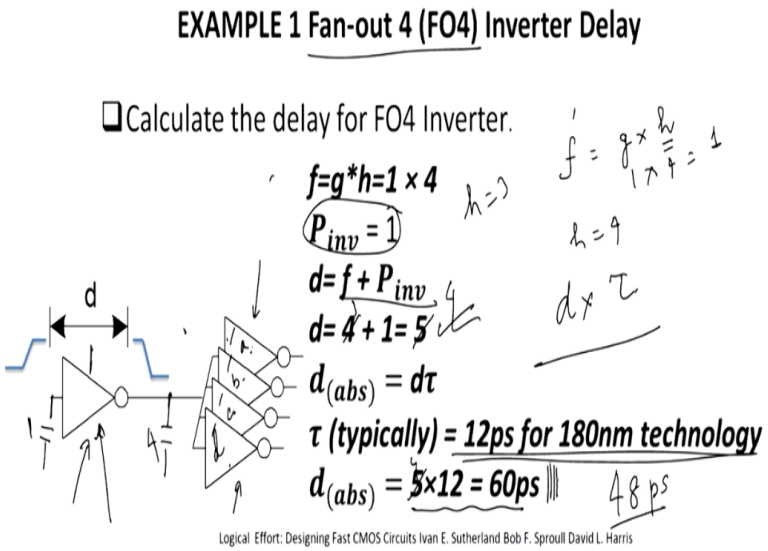


So this column is basically a logic gate this are the number of inputs which you see in front of you right and we will discuss this later on parasitic delay this is the. So you have if you have obviously you can have one input NAND gate NOR gate XOR multiplexer so you have two input you have 3 input and 1 input the general rule of thumb is that for an NAND gate is n + 2 /3 is the value of your logical effort and for a NOR gate it is 2n+ 1 divided by 3. Whereas for a multiplexer the logical effort is exactly equals to 2 independent of the number of inputs and for a XOR gate it is basically a parity checker I will have number of inputs for 2 it is 4 for 3 it is 12 so on and hence so forth right.

So this is how it is how it looks like what is parasitic delay by the way parasitic as I discussed with you is primarily the delay which is happening by virtue of the capacitances and our things parasitic with in the circuitry. So if you have an inverter you will have let us suppose inverter as got a parasitic capacitance of parasitic delay of P inverter then a NAND gate n input NAND gate will have n times the P inverter because all those n gates are putting into the output side right.

Suppose I will it clear that for all your n input NAND gate or n input NOR gate the output logic will be loaded n times so your parasitic delays will be therefore n times inverter and n times P inverter here right. If you doing a multiplex it is 2 n times P inverter because not only will have both the arms but we will also have a select rise coming into the picture so these are the typical parasitic delays which you see in front of you in terms of logical effort the understanding.

(Refer Slide Time: 13:48)



We come to a FO4 delay calculation of a FO4 delay which primarily means that I have a single delay inverter here which is this one and it is driving 4 such inverters in this case that is the reason we referred to this as a Fan out 4 FO4. FO4 primarily means that Fan out basically means that if a singled sized inverter is driving similar inverter right we define the number of inverter is driving to be as the Fan out.

So if you look very carefully inverter number 1 is driving say ABC and D inverters from an external circuitry and all are equally sized therefore we defined as this to be as a FO4 Fan out 4. Now we have just now seen that F small f basically G into H. Where G is the logical effort and H is the electrical effort which you see. Now logical for an inverter as I discussed with you is always = 1 which is 1 now why H = 4 h = 4 is very simple that the reason being that if you look at the so as we are discussing about the delay by this this is example here.

So if you have a delay so if you look at this point which is the transition between the first and second case first and second case then at this point it sees 4 equally sized inverter connected in parallel. So if I assume to be a unity here 1 and this will be exactly equal to 4 provided all the inverter ABC and D have equal loading so this will be 4 so I get 4 into 1 as = f so 1 f will be equals to 1 into 4 and then therefore F happens to be equals to 1. As I discussed in the previous slide if you remember that for an simple inverter if P inverter which you see.

(Refer Slide Time: 15:39)

Logical effort & Parasitic delay of common Gates

Logic Gate	No of Inputs				Parasitic Delay
	1	2	3	n	
Inverter	1	-	-	-	P_{inv}
NAND	-	4/3	5/3	(n+2)/3	nP_{inv}
NOR	-	5/3	7/3	(2n+1)/3	nP_{inv}
XOR(parity)	-	4	12		$4P_{inv}$
Multiplexer	-	2	2	2	$2nP_{inv}$

- Where n in parasitic delay Is no. of Inputs

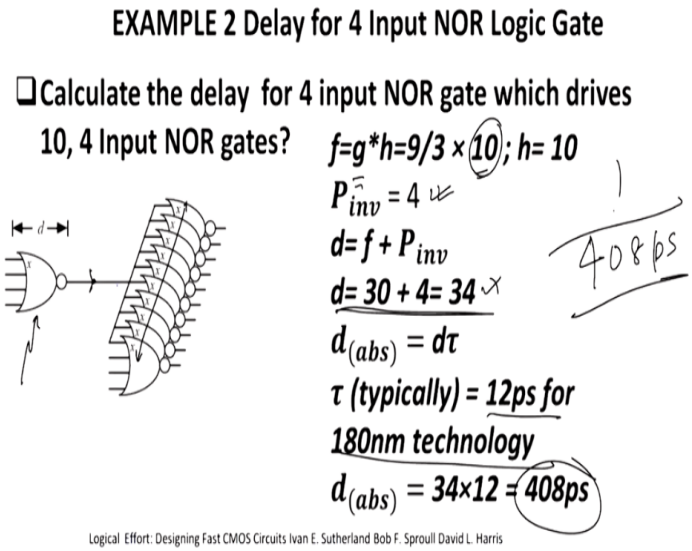
Logical Effort: Designing Fast CMOS Circuits Ivan E. Sutherland Bob F. Sproull David L. Harris

So if you look very carefully for an inverter if this is P inverter which you see as my parasitic delay right. So if you look at the parasitic delay I get P inverter = 1 so the total delay therefore will be f+p inverter f is 4+1 5. So I get the delay units of 5 delays units right I get 5 delay units right which means that the absolute delay will be noting but D into tau where tau is approximately 12 Pico second for a 180 dynamiter but this is fixed this this tau is by virtue of the fabrication technique which you are using.

So for a 0.18 micron technology this tau is typically 12 Pico second right now by virtue of the circuit you have actually seen 5 time increase in the value of tis basic value. So what I do a multiply 5 which is this one multiplied by 12 and there I get 50 Pico second as the absolute delay between input and output of a particular gate right. So this is how you calculate the fan out FO4 of a basic inverter delay or a basic inverter and this is because this is driving 4 you get 4 as your h electrical effort now if you are driving 3 I would get h = 3 so and hence so forth right.

So let us see what would have happen if h = 3 then I will get this to be equal to 3 so will get this equal to 4 and therefore I will get this to be equal to 4 and then I will get 48 Pico second so you see if inverter derives equally sized inverter either fan out is larger and if therefore we can see delay will be also larger as the fan out becomes larger and larger and the reason is the load capacitance starts to become larger and larger which reduces the delay drastically for a FO4 delay right that is the reason why we get a larger and larger delay.

(Refer Slide Time: 17:32)



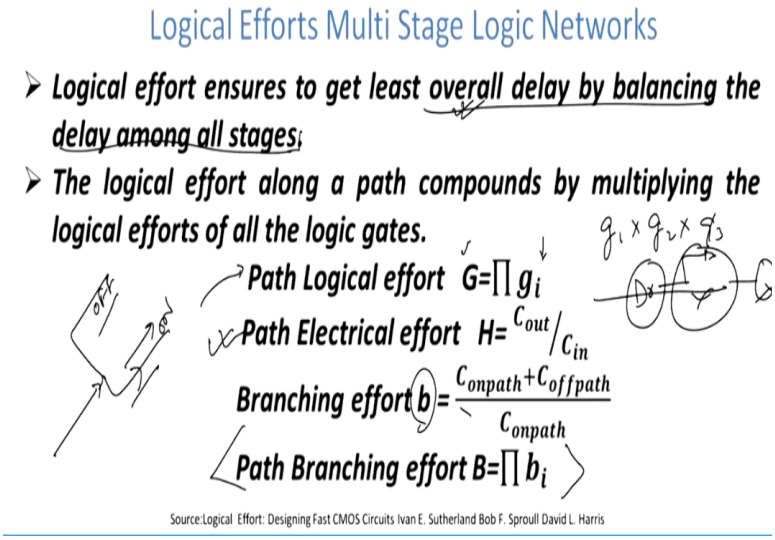
Let us look at a another example where we have a 4 input NOR gate right this is the 4 input NOR gate and this driving 10 such 4 input NOR gate so this is driving 10 such 4 into NOR gate this is the ten NOR gate available to us right. Now f again g into h since h is basically the capacitance at this point with respect to capacitance here I get h to be equals to 10 right because it is each inverter is driving each inverter is driving 10 load therefore your h= electrical effort equals to 10 g is 9/3 for a 4 input NOR gate you can find it out or even in a simple form you can find out the value then if you multiply these two together and of p inverts = 4 for a 2 input 4 input NOR.

If you add those 2 together I get d = 34 and generally it is again as I discussed with you 12 Pico second for a 180 nano meter technology multiply by 34/12 I get 408 Pico second which means that the delay between input and output for a 4 input NOR logic gate is basically 408 Pico second right so one upon that will give you the maximum frequency of operation of a 4 input NOR gate may be with 180 nano meter technology therefore even if you do not understand anything then you do not want to understand this part as such you can actually see that if your fan out is large which is which means that rather than 10 if it at 20 you automatically see the delay becomes larger and larger this is the first thing.

The second thing is if your gate is more complicated for example rather than using a let us suppose of 4 input NOR I could have used a 5 input NOR right if I would have used 5 input NOR then the g value would have been larger and therefore my overall delay will also be larger which means that more complicated the gate in terms of either a larger number of inputs or the gate becoming or the structure becoming more complicated the delay actually rises up because of that right and that is the reason we generally required to keep a lower of delay by choosing a lower more.

So many a times you must of heard that realize a logic I can use 3 input NOR right but I can also use a 4 input NOR it reduces the area but the price you pay for it is exactly the same that rather using a say may be 4 input NOR it is advisable to use 8 to input NOR to achieve the same result but doing certain manipulation using Boolean expressions and the reason being that you actually have a logical effort which is much higher and that results in a larger intrinsic delay of the effort that is what is that is the reason why we get it.

(Refer Slide Time: 20:23)

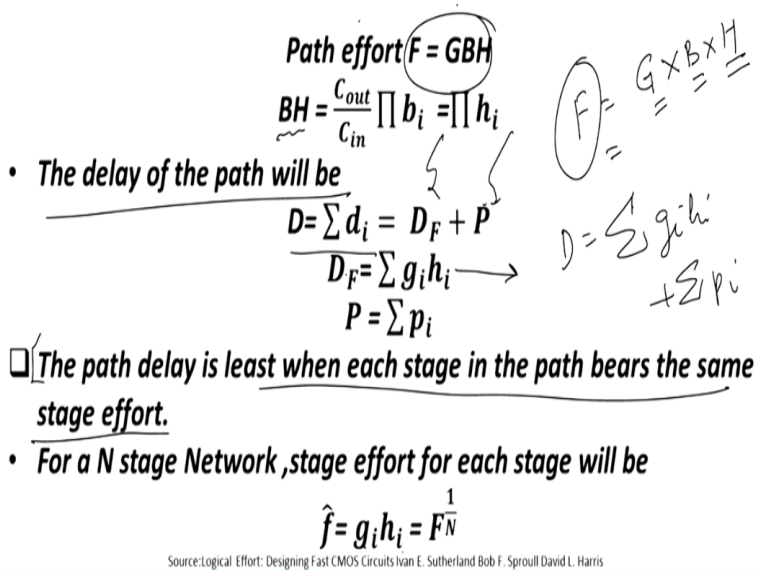


So logical effort therefore ensures that the overall delay by balancing the delay of all the stages which means that between the two stages if I am able to balance the delay I will get the minimum delay overall across the part but if I therefore multiple devices across a path then path logical effort will be nothing but the multiplication of each logical effort of each gate so this is g1 multiplied by g2 multiplied by g3. So in a path if there are say 3 transistors and then you have got something here some and then some here you multiply the logical effort of this with this and you get the total logical effort.

Similarly the path electrical effort h is nothing but C out by C in C out is the output capacitance of the end of the path and C in is the input capacitance of the primarily input which is this one right. Now branching effort is something like this that if you have a signal coming here and it is getting branched right and this is your critical path this is your critical path let us suppose then we define this to be as the on path and this to be as the off path then we define branching effort b small b to be equals to C on path C on path is total capacitances on the on path + C off path total capacitance is in the off path divided by C on path right.

And this is how you define your branching effort small b similarly if this branching is going on for a larger this thing for a large values of branching is there then the path branching effort will be nothing but the multiplication of all this branching efforts to larger extent.

(Refer Slide Time: 22:01)



Source:Logical Effort: Designing Fast CMOS Circuits Ivan E. Sutherland Bob F. Sproull David L. Harris

So we define a new term now we define the term to be as f which is basically known as the path effort and it is equals to g multiplied by b multiplied by h. So g is a total path effort total logical effort this is your electrical effort and this is your branching effort so if you multiply all three together I get f total path effort right and if you multiply b into h which is this one I get summation hi this thing. Therefore path delay of the path will be nothing but summation of individual delay which is nothing but df + parasitic delay which you see here.

We have already discussed this to be as equals to gihi so finally what I get delay to the equals to is equals to gihi+ summation of pi. Pi parasitic delay this thing right now the path I will just make a statement here the path delay is the least when each stage in the path be as the same stage effort so if you got single path and in the single path you have path delay then if the path delay is least right if multiple paths are the same stage effort right and that is pretty important which you see in front of it.

So for at n stage network stage effort for each stage will be gi into hi which is nothing but f capital f to the power 1 by n which you see here right and then what is the meaning of a stage effort as far as is concerned.

(Refer Slide Time: 23:29)

The minimum achievable delay along a path will be,

$$\hat{D} = N * F^{\frac{1}{N}} + P$$

$$\hat{h}_i = F^{\frac{1}{N}} / g_i$$

$$\hat{h}_i = \frac{C_{out-i(total)}}{C_{in-i}}$$

The capacitance seen at the input of each stage will be,

$$C_{in-i} = C_{out-i(total)} / \hat{h}_i$$

Source:Logical Effort: Designing Fast CMOS Circuits Ivan E. Sutherland Bob F. Sproull David L. Harris

The minimum achievable delay allow a path will be therefore you differentiate it and get something like this where as the capacitance C with the input is nothing but C total – C hi where hi is nothing but C out – I total divided by C input so C out I 1 C in I is basically hi bar right which you see in front of you.

(Refer Slide Time: 23:53)

Recapitulation

Term	Stage	Path
Number of stages	1	N
Logical effort	g	$G = \prod g_i$
Electrical effort	$h = \frac{C_{out}}{C_{in}}$	$H = \frac{C_{out-path}}{C_{in-path}}$
Branching effort	$b = \frac{C_{on-path} + C_{off-path}}{C_{on-path}}$	$B = \prod b_i$
Effort	$f = gh$	$F = GBH$
Effort delay	f	$D_F = \sum f_i$
Parasitic delay	p	$P = \sum p_i$
Delay	$d = f + p$	$D = \sum d_i = D_F + P$

So just to recapitulate what we dealt we did today because basically give you that if you have number of stages equals to 1 we define part 2 to be equals to 1 if you in a stage ratio logical effort is g in a path it will be multiplication of g across the whole path electrical effort is C out by C in right and we define capital h for path to be equals to C out path C in path branching effort

for a stage is C on path + C off path divided by C on path whereas branching effort for whole path is nothing but multiplication of each branching effort.

We define efforts f = g into h and for a path we define g into h for sure capital g into capital H but we also multiply that to the capital B. Capital B is the total branching effort between input and output effort delay is f and if you some effort delay across the whole network we get df we have parasitic delay and therefore we will have summation of the parasitic delay to be equal to total path delay and therefore delay d = f + p which is nothing but df + p where df is nothing but summation of fi right summation fi once you have able to find the effort delay right you will be able to find out the value of this thing.

Parasitic delay is already known to you so this we have learnt till now we have also learnt are therefore we have also learnt till now today by this module was how we calculated the value of logical delay effort and therefore given any gate input gate you should be able to calculate the logical effort from logical effort given the load capacitances you can calculate the path effort once path effort is known you already know the parasitic value of the particular gate and therefore you can be able to calculate the total delay which is equals to f + p and you can actually calculate the total delay available to you right.

This gives you a brief idea and brief idea about how you are able to achieve a relatively a low value of path delay in this design right and this is the whole motivation for doing this chapter on delay fine with these I thank you for patience here thank you