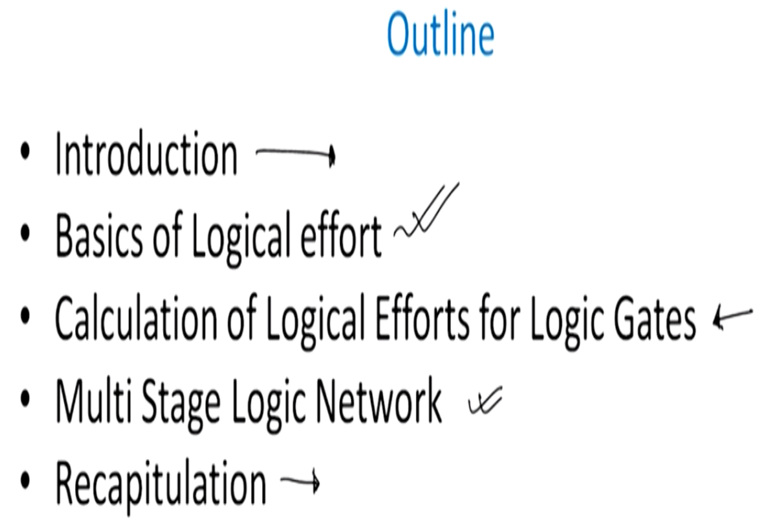
CMOS Digital VLSI Design
Prof. Sudeb Dasgupta
Department of Electronics and Communication Engineering
Indian Institute of Technology – Roorkee

Module No # 05
Lecture No # 22
Logical Efforts – I

Hello everybody welcome to NPTEL online certification course on CMOS digital VLSI design so this stage we have already finished the combinational logic design using CMOS technology any output of combinational logic why this is useful is that in a longer run when you have lot of combinational logic available to you and you need to find out the frequency of operation of the logic block. Such a delay calculation will be quite important in terms of certain the areas where you can actually reduce the delay drastically.

So the this module is on logical efforts so this is the first module in logical efforts which will be which we will be doing at this stage of time the outline of the talk will be that I will be introducing to you the logical effort what do I mean by logical effort I will explain to right.

(Refer Slide Time: 01:34)

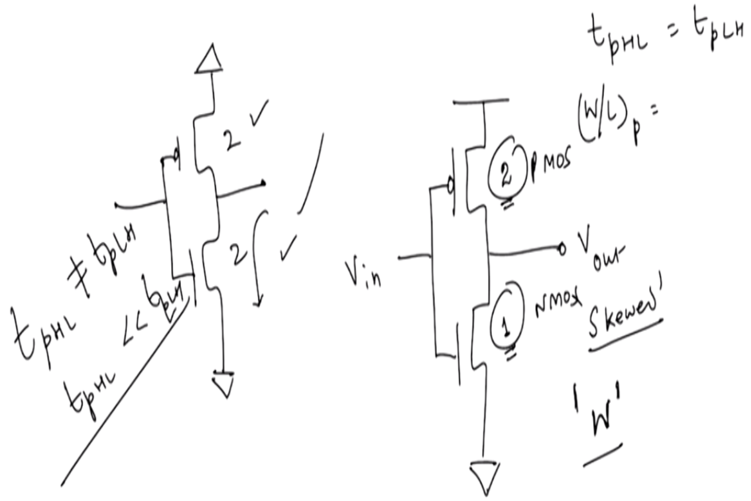


Outline

- Introduction
- Basics of Logical effort
- Calculation of Logical Efforts for Logic Gates
- Multi Stage Logic Network
- Recapitulation

So this is where we will do the introduction of logical effort then the basic concept of logical effort will be taken care of in the next part of the term then if given therefore certain gates for example ordinary gates which are available to me how can I circulate the logical effort right and then for the single stage gate. Now if you have multi stage networks when you have a cascaded networks available to you then in that case how do you calculate the logical effort. So and then we will finally recapitulating the whole issue I am concluding what we learnt as far as this module it concerned which means to which comes to a basic issue to that.

(Refer Slide Time: 02:27)



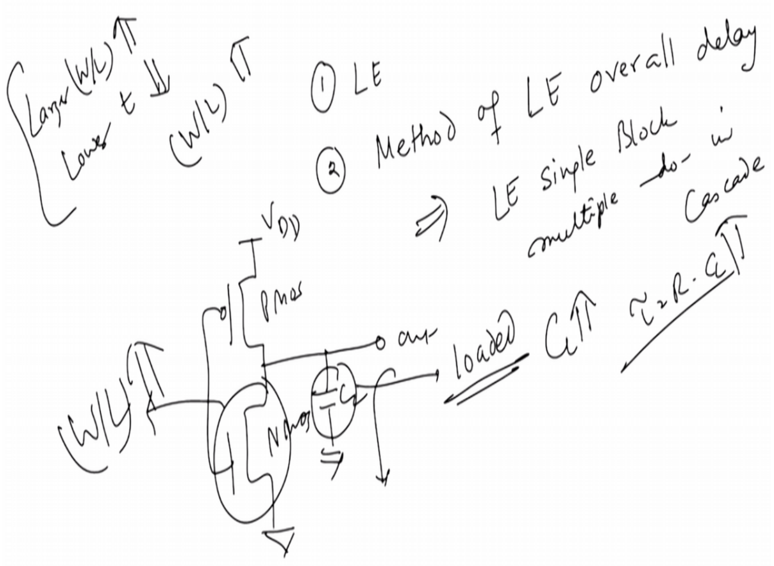
If you remember we had learnt in the our earlier modules that if you have a simple transistor CMOS transistor right CMOS transistor then if you have a MOS device and you have an NMOS here and this is your V out right and this is your V in then we have learnt that for the design to have equal delays which means that to have that tpHL = tpLH require a W/L ratio of PMOS should be approximately twice the W/L ratio of NMOS right.

So this will be effectively twice as compared to 1 if W/L ratio of NMOS is 1 you require to make the W/L ratio of PMOS which is the pull up device to be approximately = 2 and the reason was that the mobility of holes is approximately one third or one half of electrons it is pretty low and therefore to ascertain the same current in the pull up path and resistance offered in the pull up path to be equal to the pull down path you require then this delay should be this aspect ratio to be maintained right.

So you have a you therefore have a skewed design here why because your aspect ratios are equal if you would have been in aspect ratio would have been equal which means that this is rather than this two and 1 if both are actually say 2 and 2 then of course your tpHL will not be equals to tpLH and which is will be higher of course since your pull down path as got a low resistance your tp high to low will be much lower s compare do tpLH right.

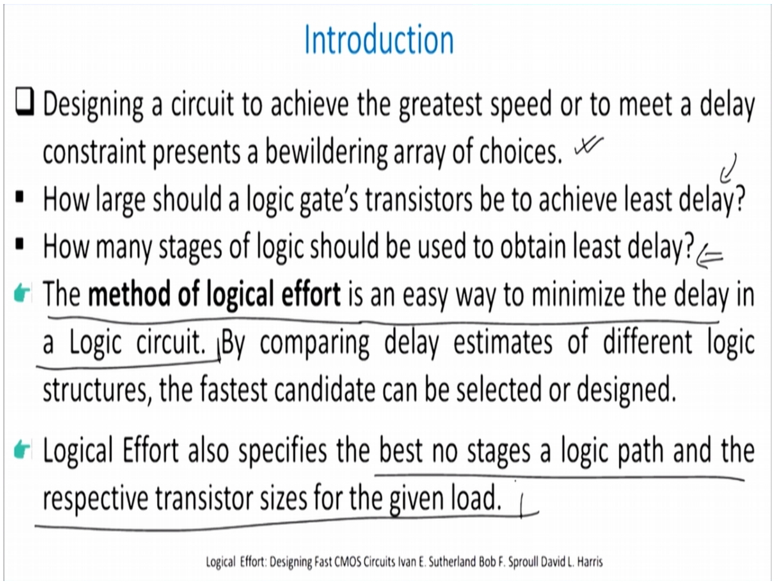
So they are not equal therefore which means that low to high and high to low transition not equal incase of in case of equally sized PMOS and NMOS but they are equal if they have got 2 is to 1 ratio of the aspect ratio available to you. Please understand we do not generally change the value of L it is already fixed by the fabricator and the technology node as a designer what thing you are having your hand is basically change in the value of W. So you change the value of W to suit your requirements as far as logical effort is concerned as far maintaining the exact same amount of delay is available to you. So let me just introduce to you why this topic came into picture and why we are actually interested to find out.

(Refer Slide Time: 04:54)



So what will be learning is basically therefore that the first thing which will be learning is basically the logical effort and then second the method of logical effort method of logical effort to find out over all delay right and therefore if I am able to find out the logical effort of a single block how can I find it for multiple block in multiple block in cascade right. So I need to find out these things in cascade so these thing have to be taken care as far as this this module is concerned.

(Refer Slide Time: 05:26)



Introduction

- ❑ Designing a circuit to achieve the greatest speed or to meet a delay constraint presents a bewildering array of choices.
- ▪ How large should a logic gate's transistors be to achieve least delay?
- ▪ How many stages of logic should be used to obtain least delay?
- ☛ The **method of logical effort** is an easy way to minimize the delay in a Logic circuit. By comparing delay estimates of different logic structures, the fastest candidate can be selected or designed.
- ☛ Logical Effort also specifies the best no stages a logic path and the respective transistor sizes for the given load.

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

Now if you let us look at the introduction as I discussed with you designing a circuit to achieve the greatest speed is basically the requirement for any good designing that means if you have a circuit available with you achieve the greatest speed which primarily means that minimum delay what hah what changes should you do in the circuitry in terms of not its architecture but in terms of it is aspect ratio so on and hence so forth so the delay is minimize within the primary input and primary output.

So that is the main reason why this course or this module is been taught now as I discussed with you that higher the higher the W/L ratio so larger W/L ratio will imply will imply a lower a lower delay right that we have already seen just now but that should have been taken in a very literal sense of it and the reason being that as you make a W/L larger and larger you also tend to load your output load capacitance I will explain to you what do I mean by that.

Let us suppose I have got a load capacitance of course load capacitance seen here now you have as I discussed with you this is my output here and this is my VDD and this is my PMOS and this is the NMOS pull down network and this is my load capacitance so suppose you want to increase the W/L ratio of NMOS you want to increase W/L ratio and you want to make it larger and larger well if you want to make it larger no problem then the current pull down current will be larger and therefore your (()) (07:09) low delay will be reduced.

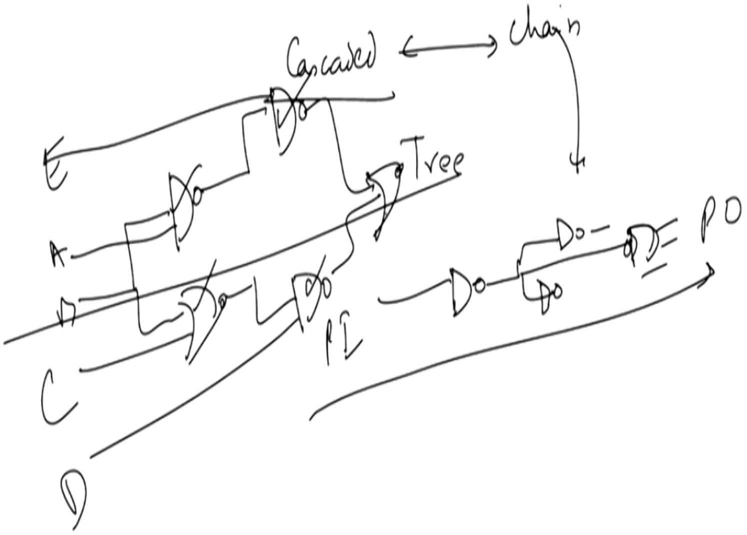
But please understand as you make it larger and larger your signal as to drive therefore a larger load as such right. So you have to very cautious right that you when you find out the value of

W/L and just saying in a generalize term that by increasing W/L will reduce your delay is not a proper. You will get a minima and then there will be a increase in the value available to you in the terms of terms delay.

So delay will be available for a very small specific minima of W/L and beyond that there might be enhancement in delay by virtue of the external loading of the capacitance right. So this CL which you see in of your might be loaded so this will be loaded as it is loaded then the CL value itself goes high and as a tau = R into CL has this will also go on increasing and therefore the delay becomes larger right.

So this is what I was decreasing saying that second stage that therefore how large the logic gate transistor be so that i get a minimum delay s it is not true that for all increase in value of W you will get a value of decrease in the value of delay right it there will be some offset because of a enhancement in the capacitance also. Now there is also concept which is third one is that means if you have a logic available with me then I can achieve the logic in two methods if you look very carefully.

(Refer Slide Time: 08:42)



And that two methods are known as you will have one method which is basically a cascaded method right cascaded method but this is basically a chain method so you will have a chain available to you can also have a tree method right which means that chain is something like this that you are able to drive a something like this right so will have chain of devices being attached here so and hence so forth this this this I have from primary this is primary input and this is my primary so I have a chain of devices can be between primary input and primary output.

It was also a tree my tree topology in which we try to divide the out the input into various sectors let us suppose and then we try to find out the delay across the various sector. So maybe it is coming from A let us suppose B and this is C and then it is close to something like this and then you have a NAND gate here this is D let us suppose and then you have a NAND gate here and then suppose this is E and so and hence so forth and then you get these together you put it to a NOR gate right.

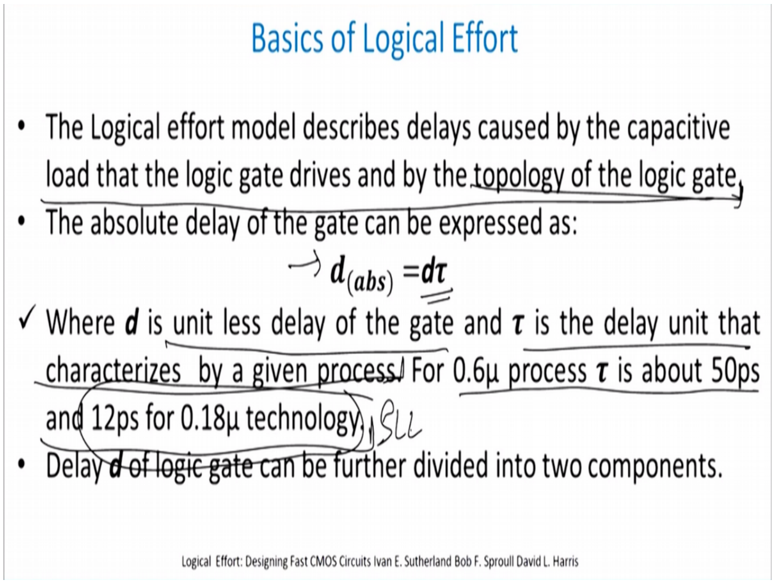
So what do you see here in the second case is that you have a tree type architecture wherein it is perfectly let us suppose assuming that both that this is NAND this is NAND and this is NAND and all are equal delay's then this is basically a symmetric network and therefore this will be no glitches which I have discussed with you with the previous turn whereas tree will have glitches but depending on the network you will have different values of delay's available right.

So how you place your logical output is also an important issues in terms of determining the total delay right. So this is third part which is there that how many stages of principal output logic which should I maintain in order to have a minimum delay between the input and output. Now therefore the logic of there the logic of therefore the logic method of logical effort we referred to as LE is an easy way to minimize the delay in a logic circuit right.

So we will try to find out is that we will try to find out the logical effort of each and every circuitry right and then see which one is minimum once we see this one is minimum we plug and play that one for your future requirement so this is what we are trying to do as far as logical effort is concerned. Now the problem is therefore that the delay will be compared with the bench mark right. So you will have a minimum bench mark available to you so i assume a equally sized inverter 2 is to 1 inverter to be a benchmark cycle.

So depending on that how complex is your gate is what is the effective loading of the gate in terms of of capacitance will determine into logical effort. Now as I therefore we discuss it the logical effort therefore specify you the number of stages and the logical and the respective transistors size for a a give load. So if define what is the load is if I define what the load is then it gives me the number of stages which is required to drive the particular load right the logical effort okay. Let me come to the basic logical effort then this is quite interesting and easily solvable also to find out the delay.

(Refer Slide Time: 11:56)



Basics of Logical Effort

- The Logical effort model describes delays caused by the capacitive load that the logic gate drives and by the topology of the logic gate.
- The absolute delay of the gate can be expressed as:

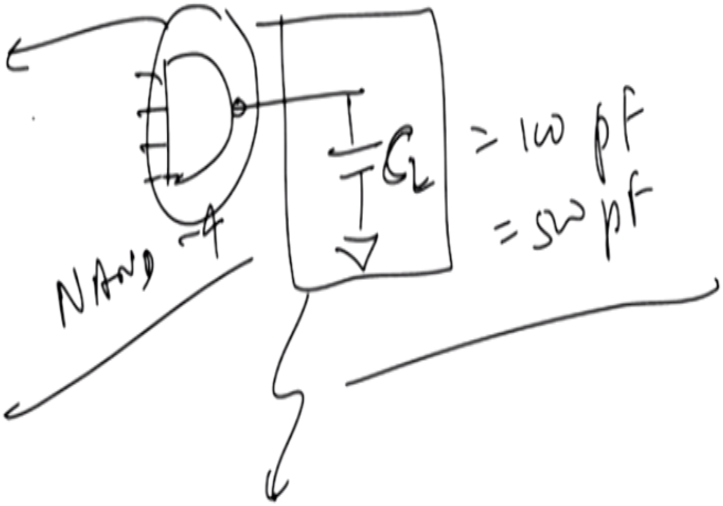
$$\rightarrow d_{(abs)} = d\tau$$

- ✓ Where $\boldsymbol{d}$ is unit less delay of the gate and $\boldsymbol{\tau}$ is the delay unit that characterizes by a given process. For 0.6µ process $\boldsymbol{\tau}$ is about 50ps and 12ps for 0.18µ technology. SLL
- Delay $\boldsymbol{d}$ of logic gate can be further divided into two components.

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

As you can see the logical effort model describes delays which is caused by capacitive load that the logic gate drives and the topology of the logic gate and I will explain to you what these what these two things are it means to say that.

(Refer Slide Time: 12:14)



If you have let us suppose you have gat right and then the gate is driving some external logic right external logic whose value is CL right then this value of CL load capacitance will actually determine how fast or how slow your this devices in a sense that external load capacitance will be responsible for either pulling the value of voltage coming to 0 right so you have initially 1 here want to go to 0 if you make CL larger it will take a larger amount of time to do that right.

So what I want to write down that the logical effort model describes delay caused by the capacitive load that the logical gate drives into the topology of the gate. So you see therefore this is basically a NAND 4 logic which is means that NAND with the 4 input logic and it is driving a CL of say 100 Pico farads right. Suddenly by some problem or some either means because of increase in W/L ratio or the NAND gate let us suppose this goes to 500 Pico farads right.

Then there is a drastic change in the output voltage time taken for the output voltage to reach the (()) (13:26) value and that is what we are talking about logical effect will help me to fin out those changes in a design wherever there is a increase in the loading or decrease in the loading the second part of the logical effort is so this is the first part which is nothing to do with the transistor itself.

So I have a transistor here right and have this is a external part of the transistor now this external part your internal part is nothing to do with what we have discussed now in the internal path as you can see it also tells me by the topology of the logic gate which means that how complicated is your logic gate right will also determine it the delay for example a 4 input NAND gate will have a very large value of input capacitance right.

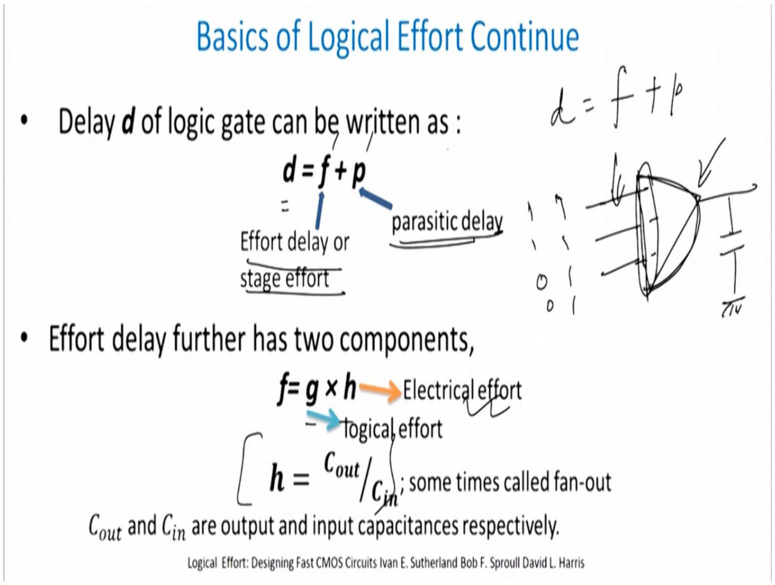
Whereas a 2 input NAND gate will relatively have a lower value of input capacitances right so therefore the price I am paying for reduce number of input is that I am trying to increase the number of transistors CMOS transistors to achieve the same result earlier. So one has to very careful in this terms that when we say that we are trying to find out the value of the output load capacitances that depends not only on the type of gate you are using but how complicated your gate is as compared to single inverter right.

So that is pretty important that one as to be very careful about so define the absolute delay of the gate as the absolute to be equals to D into Tau where D is the unit less delay of the gate and Tau is the delay unit that is characterize by given process. So then only for a 0.6 micron process Tau is about 50 Pico second and 12 Pico second for 0.1 technology.

So if you are using a 0.18 for example let us suppose a SCL technology right then you will typically get 12 Pico second as your as your delay unit right and that make sounds the sound sense that sound senses also that if the process is very complicated right the delay will also very large therefore that is because of capacitance seen by those gates are typically very large.

So this is what we have understood till now and it is pretty simple then easy to gather therefore once we have understood that the delay primarily influenced by the type of logic here you are using as well as the type of delay which we are expecting to get. Let us come to the third next topic and that is the delay of the logic gate can be further divided into two components.

(Refer Slide Time: 16:08)



Basics of Logical Effort Continue

- Delay *d* of logic gate can be written as :

$$d = f + p$$

Effort delay or stage effort

parasitic delay

- Effort delay further has two components,

$$f = g \times h$$

Electrical effort

Logical effort

$h = C_{out}/C_{in}$; some times called fan-out

C_{out} and C_{in} are output and input capacitances respectively.

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

So I will just see what is these two components are the two components which we referred to delay of a referred to is written as D = F +P where F is defined as the effort delay or stage of a and P is the parasitic delay. So we define D which is basically your logic gate delay is nothing but a some of F and P where basically P is the parasitic delay and F is the effort delay or stage effort I will explain each one of them individually before we move forward.

Parasitic delay is something like this we have already understood this is out previous discussion that let us suppose I have a MOSFET and I have got my gate say source and drain like this and then gate is coming like this. So I am assuming my Cox to be this much here right and as a result there will be an inversion layer of formed at this between point and therefore if I give a voltage

VG can say that Cox = I can say if del Cox will be cell Vgs in divided by sorry del QVgs divided by del QVgs right why?

Because if you cross multiple I get Cox Del Vgs equals to del of Q surface (()) (17:28) fine so what we have learnt therefore is that higher the value of your parasitic right if you see the parasitic here then we define d to be equal to f + p where f is the effort delay or the stage effort and p is the parasitic delay and so that is what we do not know about. Now effort due effort delay we have stage effort is again broken down into 2 components that is g into h.

H is defined as the electrical effort and g is defined as the logical effort right so g is the logical effort and h is the electrical effort. We define h is also to be equal to Cox by C in so which basically means that if I have got three input logic gate and let us suppose I have been inverted in the output side then we see under the condition that because this is let us suppose this is NAND gate let us suppose that is easier way to explain understand also.

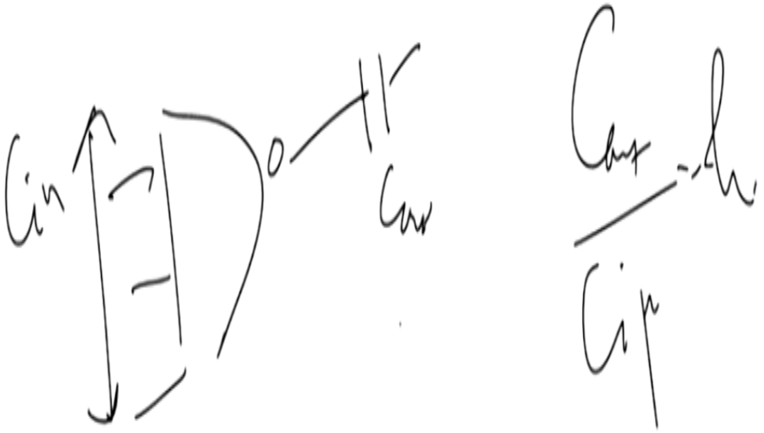
Let us suppose I have a NAND gate right then in this NAND gate let us say all your inputs are NAND gate 1 right another case is when to the input are 1 of them are 0 then the capacitance seen by the device in the first case the higher of compared to that seen by the device in the second case fine. Which effectively means that the capacitance loading on to the gate by virtue of the input is happen to be a strong function of your input itself.

So if you input varies strongly I would expect to see a large change in value of delay at this stage nothing we are saying about we are just saying that if there is a variation in the input there will be a variation in the number stages in the primary as a result we will have change in the value of your output delay right. So effort delay f is divided into g into h where g is the logical effort again and h is the logical effort which you see.

H is also defined as C out by C in right so h is defined C out by C in what is C out by C in I will explain to you let us suppose I have a NOR gate right and it drives a say for example a inverter right and I want to find out the logical effort at this particular point and the logical effort will be primarily given by the if you look at this is this thing that it is C out by C in which means that it will look at a let us suppose here it is C out right and you are giving C in here as let us say some of C1, C2 right C3 right suppose C1, C2 and C3 then we define total capacitance = C1 + C3 fine and we can actually have a large value capacitance available here right.

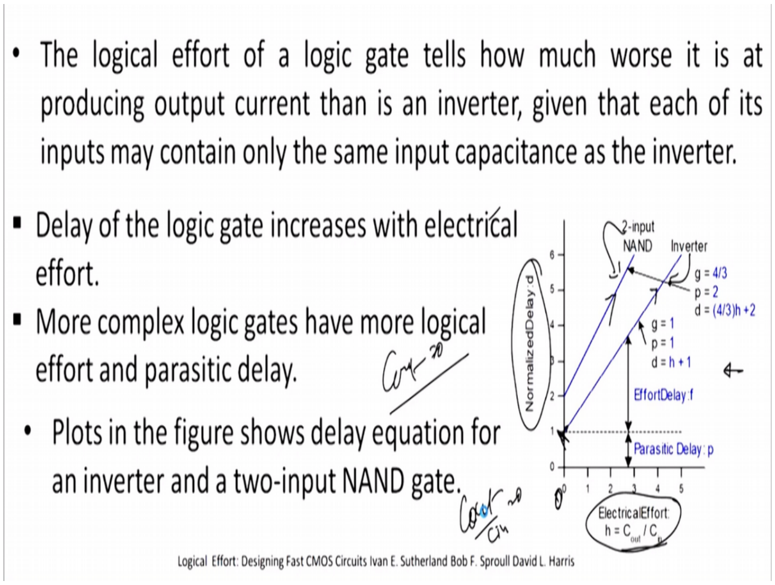
So we have understood therefore the electrical effort the electrical right so logical effort electrical effort is basically therefore C out by C in right.

(Refer Slide Time: 20:31)



C out by C in primarily meaning that if you have a device and you have C out this is C out and this is the C in value which you see this is C in value right. So it is basically C out by C in is basically H value right.

(Refer Slide Time: 20:47)



- The logical effort of a logic gate tells how much worse it is at producing output current than is an inverter, given that each of its inputs may contain only the same input capacitance as the inverter.
- Delay of the logic gate increases with electrical effort.
- More complex logic gates have more logical effort and parasitic delay.
- Plots in the figure shows delay equation for an inverter and a two-input NAND gate.

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

So you see therefore if you look at the first sentence of the slide the logical effort of logic gate tells how worst is it at producing output current than an inverter given that each of its input contains only the same input characteristic of the inverter right what do you mean to say by this? I mean to say by this is that if I have a logic gate right and I am it has been driven by a external data source then for every input which goes into the into the AND logic right every input which it goes your capacitance value starts to change and as a result that logical effort becomes a strong function of the input voltage available to you first thing.

The second thing is how much difficult is your current design as compared to your previous design which was just previous to the this one how is it different how is it worse that's compared to previous so I have a first design which is the very basic back of the envelope assuming that to be giving you the best output delay what we are looking currently here is that if there is change in the output change in the input profile then how it is difficult to produce the same current as your inverter was minimum size inverter was giving you a current.

So that is what is all about logic gate as such now delay of the logic gate obviously increase with the electrical effort now why this is so I think i need not go into details of this one because electrical effort if you remember here is C out by C in right is C out by C in now C out by C in if you look very carefully and if you find out there so if it more complex your C out by C will be also so larger right your output capacitance will be also very strong function of an input not only that C out by C in will have a substantial value and therefore we will be position that your delay will be typically large.

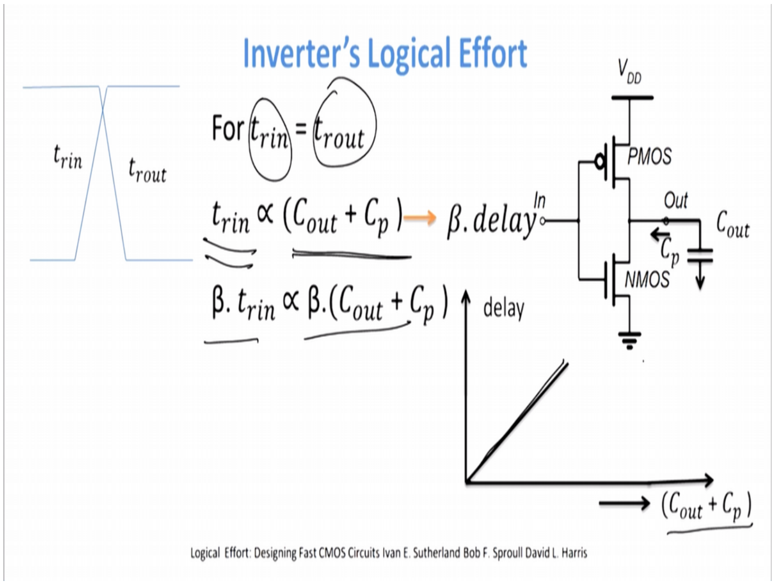
Now more complex a gate is as I discussed with you just now more will be the logical effect available to you right which means that for if I take a 2 input NAND and I take 5 input NAND obviously a 5 input NAND will give you a larger logical effort as compared to 2 input NAN. So one has to be very careful that a logical effort is strong function of the architecture of the system which we are using.

Whereas parasitic delay and other delay are primarily look in the device point of you in the plot which you see here on the right hand if you look at the plot here. It is basically a plot of normalize which in the Y axis right as compared the logical effort which is on the X axis so this logical effort is given as C out by C in means the output capacitance by input capacitance as you therefore increase C out by C in both my inverter and my NAND gate logic function both starts to increase and you can understand why?

The reason being that for even for a inverter if you are go on increasing the parasitic delay you go on adding delay at a linear fashion to the input and therefore it starts to move ahead at a faster phase right and that is what is happening if you look very carefully. Now more complex logic gates have more logic efforts therefore larger parasitic delay we get the display this later on so this is a basically a first one is basically a 2 input NAND gate this one is the 2 input NAND gate right and this basically an inverter right.

Now these access where they cut the Y access that is basically with logical effort 0 primarily means by C out = 0 C out by C in = 0 implies that my C out = 0 which means that if there is no output capacitance of available with you then I will be using C out by C in from h = 0 for every other value I will have a h available with me which will be fed directly to the equation to get electrical effort.

As you can see here 2 input NAND gate will have a will have a logical effort 4/3 with the part of a effort of 2 and D = 4/3 into h+2. So I just have a look into into the fact that when you can design a delay or when you can design a logic. Now you see here therefore at 2 input NAND gate is primarily more complicated in terms of structures as compared to simple 2 input NOR or even a sorry a single input inverter right because NAND gate and NOR gate all derive from CMOS inverter only and therefore typically these transistor which is basically CMOS based amplifiers as well as for CMOS based design seems to be very difficult as compared to a simple inverter. Now so this is what you have learnt let me come to the inverter logical effort therefore.
(Refer Slide Time: 25:59)



Inverter's Logical Effort

For $t_{rin} = t_{rout}$

$t_{rin} \propto (C_{out} + C_p) \rightarrow \beta.delay$

$\beta.t_{rin} \propto \beta.(C_{out} + C_p)$

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

So assuming that your input rise time which is basically my this one trin is the input and tr out means rise. So trin basically means that the current is coming into the system right and that time the value of current coming from 10 to 90 % is basically varying from 10 to 90% is basically my input current and when the output side the current of the voltage is falling from higher value to less than 10 volts or 10 milli amps we get what is known as a r out delay out right fine this is what we get.

Now input therefore if you look it very carefully input impedance or input resistance or input time taken is primarily nothing proportional to because nothing proportional to because it is proportional to C out + CP right my CP is basically the intrinsic parasitic capacitances and that is equal to beta times delay fine. So be very sure what is beta what is delay is primarily the delay available to you by virtue of both the things and beta is factor which is governed by the equations.

So therefore if trin is proportional to this whole quantity we can safely write down beta times trin is proportional to beta times C out + Cp right we just simply multiply both sides well as Cp right okay once you do that you end up having a delay versus C out by Cpl linear plot so you see a linear plot here which means that C out + Cp goes on increasing the delay goes on increasing linearly right and that is expected also because as C out + Cp.

Cp is the parasitic delay and C out is the load capacitances which you have put here is increasing so basically when you increase the aspect ratio of PMOS and NMOS you increase the parasitic delay and therefore your over delay starts to become larger and larger in this case right and that is what inverter logical effort is all about now what we will do is will do a logical effort calculation and we can do it later on but me recapitulate what we done in this module.

The what we did in this module was primarily have a look at what is known as logical effort on what factor did the logical effort work and what are the what is this influence on the overall delay of the system right in subsequent module we will be discussing the method of calculating logical effort and then maybe we take an example to design logical effort fine thank you very much for your understanding thank you very much.