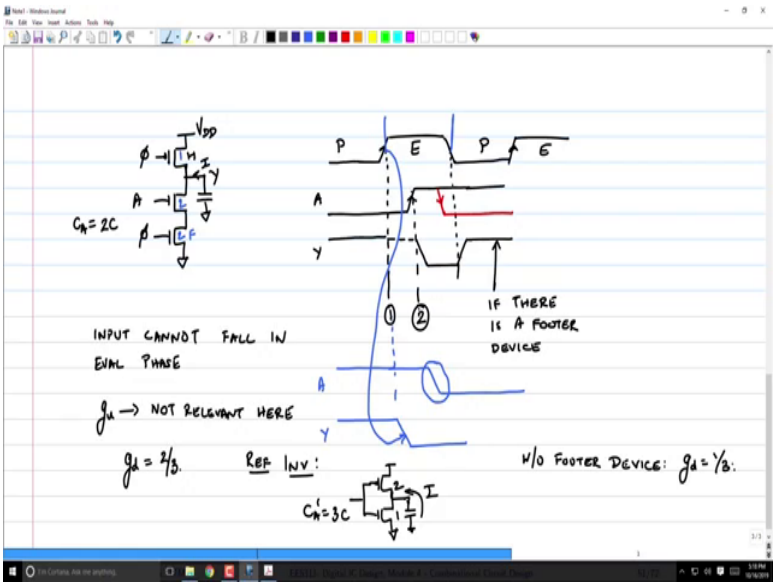
Digital IC Design
Prof. Janakiraman Viraraghavan
Department of Electrical Engineering
Indian Institute of Technology, Madras

Lecture – 51
Domino Logic and Weak Keepers

(Refer Slide Time: 00:14)



So now, let us come to the sizing. If I have a footer device what should be the size of my NMOS transistors? Yeah.

Student: (Refer Time: 00:25).

Exactly this is right. So, I will say it 2 2, header device should be? 1, I do not need to size that up, because remember my header device is not playing any part in the logic evaluation. Evaluate phase is all is only place, where logic evaluation happens.

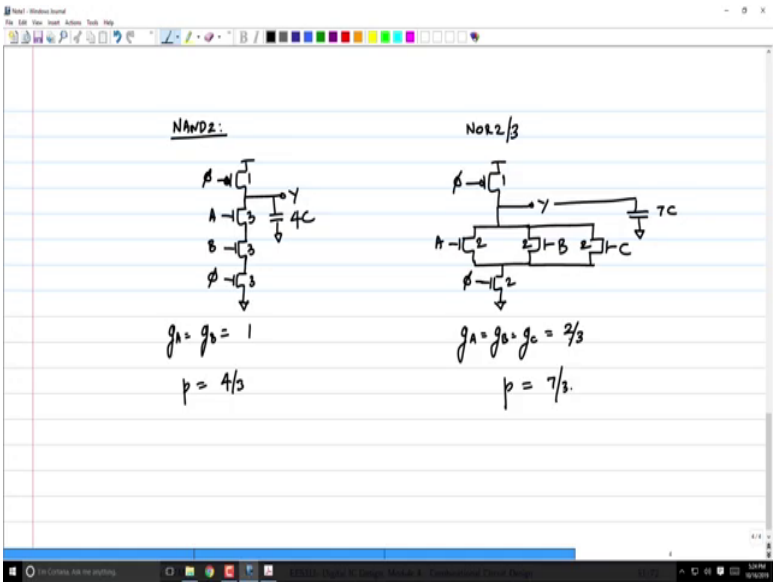
So, now first question before we even go head and evaluate logical effort. What is the pull up logical effort of this circuit? It is an irrelevant question, because in the precharge phase the output has been pulled high already, logical effort is some estimate of how long it will take for the output to go high or low right. But, for the output to be high in a dynamic circuit, you have to only look at what happens in this evaluate phase. By the time you are at the evaluate phase, the precharge is already take in it high.

So, therefore, there is no delay in taking the output to a logic high and there for logic pull of logical effort is irrelevant in this case, which not 0, it is irreverent. So, g u right not relevant here. So, what about g d, pull down logical effort? You got to take the ratio of input gate capacitance to the gate capacitance of a symmetric static CMOS inverter. You have to construct that inverter right. So, what is the gate capacitance of this guy C A? 2 C exactly. Now, what is the pull down strength of this gate or how do I construct the reference inverter for pull down logical effort?

So, if I have a capacitor here, what is the current coming out to the load capacitor during discharge? I, this is I and therefore, I have to look at the an inverter which can provide the same current as that. So, therefore, this has to be I, which means this has to be how much? 1 if this is one what is a PMOS? 2. Therefore, C A here, C A prime is how much? 3 C therefore, what is the pull down logical effort? 2 by 3. Clearly, this is a number that is less than 1. So, it is faster than your static CMOS inverter whose logical effort is 1.

Now, if I did not have the footer device what would be my logical effort? Right. Let say I could impose and additional constraint apart from the fact that the input cannot fall right, I also impose a constraint that in the precharge phase I will ensure that the input is not high let say, then I can do away with the footer device. So, what will be the logical effort in that case? 1 by 3 exactly: so without footer device g d 1 by 3.

(Refer Slide Time: 05:04)



Now, how do I construct a NAND 2 gate? Yeah.

Student: (Refer Time: 05:13).

Yeah yes.

Student: (Refer Time: 05:26).

Yeah.

Student: (Refer Time: 05:28).

Correct.

Student: (Refer Time: 05:31).

Exactly, whether it is possible like an memories circuit it is possible. For example, I know that during this entire phase I am going to my word line only has to turn on right, then the input just goes high, I know is going remain high for ever after that. Those kind of things, I can do with this kind of a circle.

What about NAND 2? The NORs are the mast useful will, will come to that. But how does an NAND 2 gate with a footer device look like? My clock, again the clock. So, what is this, what is size of my PMOS transistor first? 1 NMOS 3 right, 3. What is a logical effort now? This is A B. What is a logical effort? g A equal to g B 3 C by 3 C correct, you get 1.

So, a NAND 2 get which has 4 by 3 is now down to 1 clear. Now, what about NOR 2 logic? Again, one footer device, I mean header device connected to the clock output. Now, I need to have parallel logic here A B, this is my normal NOR gate, if I wanted a dynamic NOR gate without the footer device it would be this.

Now, I add a footer device sorry is an NMOS again connected to the clock. So, what is size of the transistors now? Sorry, PMOS is how much? 1, NMOS 2 and 2. What if I made this NOR 3? Same thing right, C right. NOR 3, then this would be again 2. So, what is the logical effort now?

g A equal to g B equal to g C; how much is it? 2 by 3. What is it 2 by 3 irrespective to a number of the inputs of the NOR gate? It is that is the biggest advantage of a dynamic NOR gate. What would otherwise scale as 2 N plus 1 logical effort is now, simply going to be 2 by 3 a fixed number irrespective was number of inputs clear. Any questions here? Yeah.

Student: (Refer Time: 09:13).

Pseudo.

Student (Refer Time: 09:23).

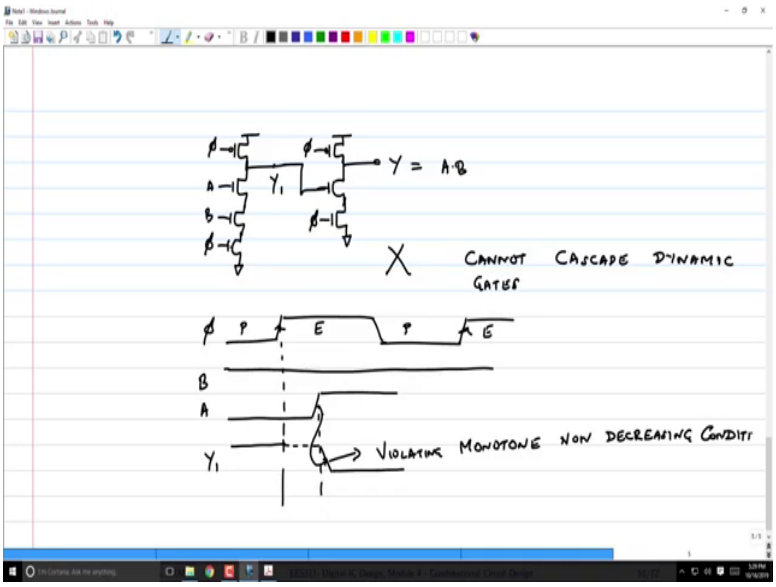
Correct, there NAND will be constant if your talking of if I do was pseudo PMOS right, where I put an NMOS transistor below. And the PMOS network remains as it is yeah, there of course, the NAND will basically have same logical effort, your right correct ok.

So, clearly this logical effort is lesser and what about the parasitic effort, let say that parasitic effort I am just going to count the diffusion capacitance. What is the diffusion capacitance on the output? 4 4 C right. So, what is the parasitic effort here? 4 by 3. What about here? This is 7 C. Now, what if what happens if I increase a number of inputs to the NOR gate.

So, that clearly is going to go a linearly with the number of input parasitic delay does depend on a number of inputs, it is only logical effort that is less, but remain remember, parasitic effort is a constant number that adds to that delay optimization problem, it is the logical effort that the ultimate the determine great sizing therefore, logical effort is more important than the parasitic effort and it that is constant that is a big advantage ok.

So, therefore, here parasitic effort will simply be how much 7 by 3 and of course, remember I have not indicated explicitly, these are all only for pull down as I g A g B g C it is actually g A d g B d g C d right. Pull up is irrelevant in this case.

(Refer Slide Time: 11:18)



Now, great I have a dynamic inverter, I have a dynamic NOR gate, I have I can make any dynamic circuit that I want. So, next what do I want to do have want to cascade these gates. So, I take a let say I want to build a dynamic AND gate ok. What do I do am going to connect a dynamic NAND gate A B phi right and am going to connect it to a the clock here, this is my inverter, this is my footer clock Y, I want Y equal to A into B, this is what I wanted to behave like, will this circuit work?

So, what are the constraints on the input A and B? It can only it can only raise right, it cannot fall right, that is a monotonicity condition. The signal can be monotone non-decreasing, it means it can remains flat or it can raise, it cannot fall right, that is a constraint. So, since it is a dynamic NAND, I would ensure that A and B do not fall during the evaluate phase.

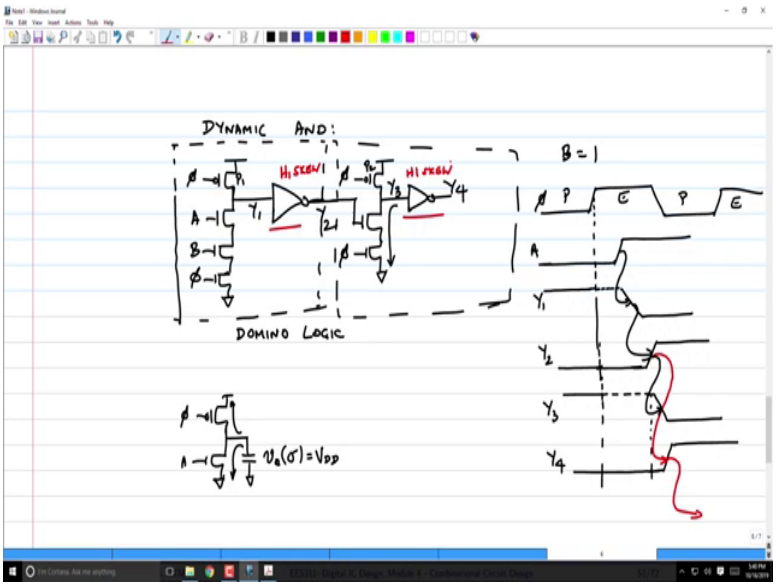
What happens to the inverter? Let say that both A and B my clock is here, let us say let us say that my know for simplicity my input B is high al through and my input A went high like this ok. This is Y 1 and that is Y. What happens to Y 1? So, again let us plot this very clearly and carefully Y 1 will be, this is precharge phase, this is evaluate phase precharge evaluate right.

Y 1 is logic high in the precharge phase, then input A is 0. So, therefore, if output will now, float high impedance still be input a raises, then this will fall right. This is the falling transition that is causing this output to fall right, that is what that arrow means this arrow here, basically indicates that, because that signal was rising that cause that output to fall ok.

So, Y 1 has now what is it done? It has violated the monotonicity condition on my inverter though my inputs A and B have, you know adhered to that condition Y 1 has violated the kind is very obvious, because the gates are inverting in nature. If I am monotone increasing, then the output is going to be monotone decreasing and therefore, I cannot cascade to dynamic gates just like that right.

This circuit will not work right and of course, this one is violating monotone non decreasing condition. So, therefore, how do I even use these gates? I have to do something, because if I cannot cascade gates the completely useless for me.

(Refer Slide Time: 15:46)



So, I for the only way to overcome this problem is to put as static inverter between two dynamic gates right. So, the same thing I can now get a dynamic AND as by building my dynamic NAND first A and B now here, instead of connecting a dynamic inverter I will connect a static CMOS inverter and then I can connect it to any dynamic gate after that.

Why is that? Because now, if A and B are monotone increasing output Y 1 will be monotone decreasing, but then the output Y here, will again be monotone increasing which is what is going to go to the next dynamic gate right. So, this is basically called domino logic cascading a static CMOS inverter with a dynamic gate is called domino logic, you will see why, domino logic ok. So, let us look at this for example, I have this inverter and am going to connect it to another dynamic inverter here and there I am going to connect another static CMOS inverter and I will see what the output here is.

So, this will be Y 2 Y 3 Y 4 right. So, remember that this is my gate this is one unit and this is another unit. So, these are the my two domino logic gates domino AND gate domino buffer ok. So, let us just a draw a way from it is a simple case here phi let us assume B is equal to 1 always, let us assume ok. I am not drawing that B equal to 1, A is going monotone increase in somewhere here what will happen to Y 1?

Y 1 will float first when the clock turns off right, I mean initially in precharge of course, is pulled high, then it will float and then it will fall, this will cause the transition. What happens to Y 2 in the precharge phase Y 1 is what high. So, Y 2 will be low is it floating low or driven low, is driven low in the next between the clock and till the input rises is Y 2 floating or is it driven to some logic, it is driven.

So, Y 2, because it is output over static CMOS inverter will never float. So, this will basically be 0 till here. When Y 1 falls this will raise, this will cause the transition right. Now, what happens to Y 3 and Y 4, let us plot these two again carefully yeah.

Student: (Refer Time: 20:18).

No, I agree, but like he said right, there is one pull up path see now, there is a leakage path to V DD also PMOS is half, NMOS is also half. So, there is one pull up path to V DD pull down path to ground. So, these two will cause it to come and settle slowly at some point, it cannot go cannot go below that, you get the point it is floating yes. So, it cannot be at V DD always, but what is a final value final value is there is on half PMOS, there is on half NMOS between V DD and the ground.

So, these two leakage currents have to be the same, because it is series part. Here ok, multiple NMOS and suppose, I assume a trivial case assume the simpler case where there is a PMOS and there will only without a footer A. So, what is happening is there is a capacitor here which is floating at V DD right V C of V C of 0 minus now, there is a pull up path to V DD you now that through leakage of PMOS, then there is a leakage to NMOS. These two are

going to fight and eventual value will actually not come down all the way to 0. It will come to a intermediate value between 0 and V DD, depending on these two leakage currents fighting.

Student: (Refer Time: 22:01).

Yeah.

Student: (Refer Time: 22:03).

Correct.

Student: (Refer Time: 22:07).

So, you are right, if it goes below v i h it is a problem; however, generally at the speeds that we operate had the clock is not going to be very-very slow it cannot be 1 hertz 1 hertz is such a you know limiting case I gave as an example usually, you operate this a gigahertz or megahertz, it cannot discharge that first because leakage is such a small current. But, your right that, because of this I cannot operate a dynamic circuit at any frequency static CMOS logic. I can operate at any frequency, it will work dynamic logic it cannot work, because of this issue anyway.

Now, let us float Y 3 Y 3 in the precharge phase, this is precharge evaluate, precharge evaluate. In the precharge phase Y 3 is now going to be yeah, just look at Y 3, this is Y 3 it is going to be driven high to the precharge transistor. Let me call this P 1 P 2 it will be driven high through the precharge transistor P 2 and therefore, it is going to be high. Now, what will happen after precharge turns off and Y 2 is not a turned on it will float.

So, here this will basically float at V DD, the moment Y 2 turns on, it will cause Y 3 to discharge like this, correct and therefore, Y 3 will go low and this is the transition that is possibly. What about Y 4? Y 4 again is a static CMOS output so, it can never float. When Y 3 is high, it will be low all through till here, Y 4 will just be low, then once my Y 3 falls, this

will go high. Now, you look at what is happening to Y 2 and Y 4. Now, if I cascade many such gates, you have Y 2 Y 4 Y 6 and so on basically, just like a domino, the outputs will keep sort of rising or falling which we look at Y 3, they will all fall.

So, it is just like a domino you nock 1 by half and everything will do that is why is called domino logic later ok. Y 4 we output of the static CMOS logic will keep raising ok. Any questions here, yeah.

Student (Refer Time: 25:16).

True, true, true, see therefore, the moment I start putting making a circuit, then there is a spect that has to be met. It is not how many you have to put you have to size these so, that they meet a timing constraint, I can put any number as long as I can size it, they can meet a timing constraint.

Student: Early if (Refer Time: 25:45).

Yeah.

Student: we have a gap of (Refer Time: 25:48) 1 NAD 2.

Correct.

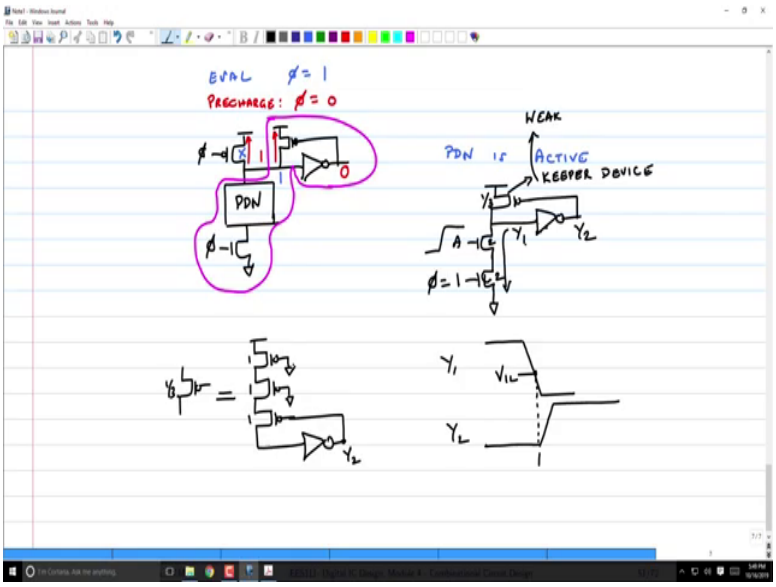
Student: Now, we are only having (Refer Time: 25:52).

True absolutely, but the logical effort is much lower there. So, therefore, it is still sort of ok, you are right ok. Before, I proceed what kind of inverter should this, inverter be these two; symmetric static CMOS or what is a constraint on Y 1 and Y 2 and Y 3?

Exactly, Y 1 and Y 3 can only fall therefore, Y 2 and Y 4 can only rise, which means all I have to do is make these two as HI SKEW. So, I will gain even on the on the logical effort,

because on the HI SKEW that was what 5 by 6 is it what was it I do not remember 5 by 6 or 2 by 3 or something like that correct. So, logical effort would be less than one even for those guys.

(Refer Slide Time: 27:12)



So just to address the question that you know he is been worried about this output falling for a while right and it is indeed a concern. So, how do you get rid of this problem? So, way you do it is anyway since, am going to cascade one inverter after the dynamic gate, what I will do is I will have my clock circuit here, I will implement whatever pull down network depending on the circuit of the logic that I want to implement, then I want to put a HI SKEW static CMOS inverter here right.

Now, I will just put another PMOS like this. What is the advantage? When the clock is in precharge phase, this output right during precharge clock equal to 0, my output of the

dynamic gate is going to be 1 therefore, the output of my inverter will be 0; that means, this PMOS will be turned on right and therefore, it is going to simply pull this output along with this guy. Both this guys will pull it up to logic 1 ok.

Now, I go to the evaluate phase EVAL phi equal to 1, what happens this guy is cutoff. Now, until the pull down network turns on, this output earlier use to float, the output of my dynamic gate use to float, but now, because there is this PMOS transistor which is cross coupled right, the output here is floating at 1 now, but the inverter is inverting that logic and causing the PMOS to still pull it up to 1. So, the output now never floats.

The output will always be at will be driven to logic 1, even in the evaluate phase. So, now, technically I can slow down my clock how much hour I want, because its being driven, only constraint is when my pull down network turned on it has to this strong enough overcome that it P PMOS that is pulling it up right. So, in the evaluate phase, when the pull down network is actived ok.

Let us just look at this part of the circuit. Let me look at this part of the circuit ok, I am going to have a PMOS that now, I am moving to have a PMOS that is driven a let us assume that am implementing and inverter in the evalue phase, this is one the footer device is turned on phi equal to 1. This output is going through a static I mean hi skew inverter and driving this.

So, let us say that this input went from 0 to 1 a went from 0 to 1 when it is 0 my output Y Y 1 was driven high through this PMOS through the keeper, this is called a keeper device, but when my input A goes high see, what is happening there is a path which is trying to pulled this down through the NMOS network, there is a path in the PMOS network which is trying to pull it up right. So, there is a fight.

This Y 1 you have to size the devices so that the NMOS network can overpower that PMOS network, which means as long as y 1 falls below V I L of that inverter that is enough for you, because when Y 1 goes below V I L of the inverter, the inverter will trip and flip. So, when Y 1, let say Y 1 was like this and it went below, some logic and let say this is V I L add this

point the inverter will trip and the output will go to logic 1 there, Y 2 so, this is Y 1, this is Y 2.

The moment it crosses V I L Y 2, which was basically 0 will now, flip and go to logic high and there by turning that keeper device off. This is very-very similar to what we did not C V S A logic we put up cross coupled thing and that output basically turned off that keeper right. The guy which was a competing for with the pull down network similarly here, this V keeper will get turned off, because the pull down network is strong enough to overcome the keeper device and therefore, this is not just a keeper, it has to be a weak keeper ok.

So, if this is size 2 and 2, then you would want to size this to may be 1 by 3 or something like that and is the typical number, the idea for weak keeper is instead of allowing it to float you weakly hold it up, that is enough, then I can slow down my clock even if there is some noise coupling nothing will happen right.

Question is can I make a transistor with size 1 by 3, for logical effort discussion it was ok, because you know whether it is 1 by 3 or 1 it does not matter as long relative to sizes are ok. So, it is fine, but in reality if I want transistor with 1 by 3 and I start possible, then what do I do? Then you have to use the idea that you can put three transistors in series right.

Each of size 1 1 1, this will effectively be like a single PMOS transistor which size 1 by 2. This is how you make a weak keeper device. Now, the problem is you see the loading on this guy. Ultimately, act to you know optimize for delay the loading on this node Y 2 is too much, because it is now, is going 3 C capacitance right.

So, what you do is you simply break this and say well I will connect only this, these to as it is ground. As long as one transistor in that sees its part of cutoff it is enough right. Here, I am reducing the capacitance, but also achieving my weak keeper to be 1 by 3 kind of size ok. This is the idea behind a weak keeper. So, I urge go back and redraw this timing diagram that you have here, if I also had a weak keeper, all this floating notes will vanish. They will all become driven high values, clear ok.