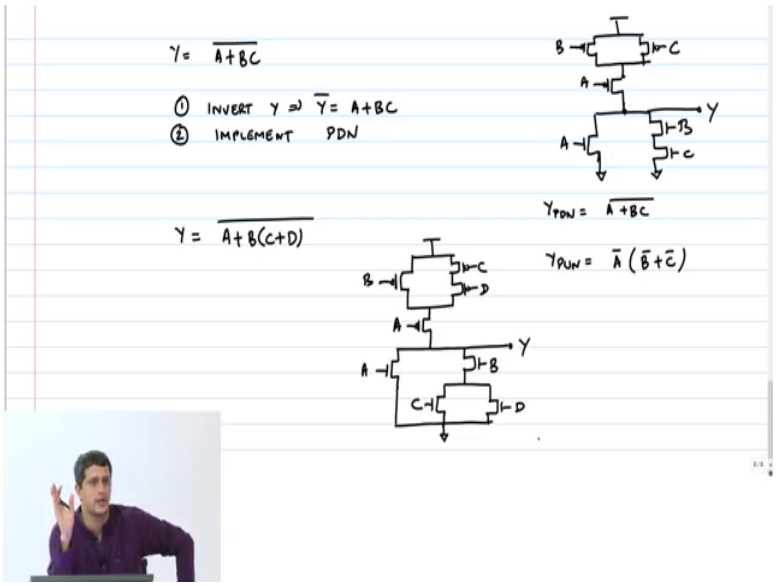
Digital IC Design
Prof. Janakiraman Viraraghavan
Department of Electrical Engineering
Indian Institute of Technology, Madras

Lecture – 34
Implementing Any Boolean Logic Function: Examples. Gate sizing

(Refer Slide Time: 00:20).



You can do exactly what I told you later simpler thing, let me take another example. Let us say I want to implement A plus BC whole bar, ok. So, first step invert function. Invert Y implies Y bar is A plus BC, ok. 2, implement sorry pull down network, ok. We will first do this step I have A plus B C how do I implement this.

Student: A in parallel.

A in parallel with.

Student: B and C.

B and C; A in parallel with B and C this is my output. Now to get the pull up network just construct the dual of this network make every series parallel every parallel series, right so, I have B and C in series, right. I need to make that in parallel and then make it in series with A that is the way you can construct the dual of the network. So, I will simply say A B and C, clear.

So, if you look at the function on the pull down network Y PDN is what, A plus B C whole bar because its going down to 0. The Y pull up network function is what, A bar plus A bar sorry, A bar into B bar plus C bar right, note that these two are Boolean equivalents. De Morgan's law I can take the compliment inside and make AND, OR and these two are the same ok. This is your check eventually both of these things should be the same. Yes.

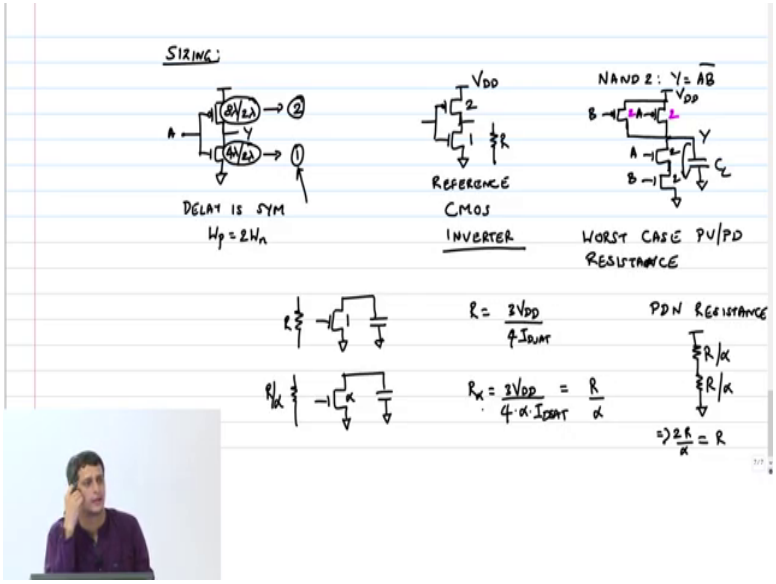
Student: (Refer Time: 03:11).

Absolutely; so, there are certain cases where these things are equivalent where logically it is to put 1 on top 1 below. I will tell you how to resolve this once we get into timing because timing wise they are not the same delay wise, any other question on implementing this logic. So, well let us do one more example which is just slightly more complex A plus B into C plus d whole bar. Can you implement this? Is this the pull down network yeah; what about the pull up? Again, I have to have A now in series with the whole thing right in series with B has to now be in.

Student: Parallel.

Parallel with C and D in series; B C and D, clear. So, with this systematic method you can construct any arbitrary Boolean function whether it will be an efficient or not we have to see later, but you can construct any static CMOS logic in this manner, ok.

(Refer Slide Time: 05:32)



So, now let us move on to the question of sizing, ok. So, we said that when we have CMOS inverter we said that the W right for example, this would be 4 lambda by 2 lambda W by L, this would be 8 lambda by 2 lambda W by L, ok A and Y. And, why did we have this twice so, that the pull up delay and pull down delay are same right, there delay is symmetric that is the reason we made W p equals 2 W n, ok.

Now, going forward since the length is always going to be 2 lambda right and the inverters with right will also be fixed by the technology parameters and so on. And in the sense depending on the technology, the designer will say by minimum width that I can use for the

inverter is let us say 4 lambda or 8 lambda or whatever. We are going to abstract this whole thing out and simply call it 1 and this will therefore become 2.

So, from now onwards I will just write a CMOS inverter like this, I will say 1 and 2, you must interpret it as the width alone is doubled for the PMOS transistor, ok. And this is what I will also call a reference CMOS sorry, CMOS inverter right, this is my reference CMOS inverter because all the gates that build from here on will be sized with reference to this inverter, ok.

So, let us say now I have a NAND 2 gate Y equal to A B bar, ok. So, what I am going to do is, I want to put two NMOS transistors in series A B. What will be the pull up network? 2 PMOS transistors in parallel, V DD A and B and this is my output Y. So, with reference to this CMOS inverter I now need to size these transistors in my NAND gate, ok. So, for example, one possible thing is to do the same thing I will say 1, 1, 2, 2, right. We understand why we did 1 2, so, that the pull up and pull down delay is symmetric. So, I can do 1, 1 and 2, 2 right.

Now, let us examine what happens to the worst case pull up or pull down resistance that is the primary criterion that we are going to pick in order to size these gates. So, we will look at worst case pull up or pull down, ok. Now we did this earlier, ok. So, I may say resistance for what resistance for delay, let us assume that a capacitor here is being discharged it has V DD initially and that is now being discharged to this pull down stack, right.

I want to look at the worst case resistance that will be offered by that stack right and ensure some constraint on that, ok. So, first let us look at when we had a single transistor with size 1 unit width 1 right as defined here by the way this one is as defined here, right. Then the resistance we calculated for this NMOS was some part I am going to rub that R equivalent N or whatever I am not going to call it R, ok.

Now, if and unit NMOS has resistance R what will be the resistance of an NMOS with twice the width. R by 2 exactly so Y is that lets look at that R was basically 3 V DD by 4 I DSAT, right. This is what the expression we got and now if my width goes up by 2 or by factor alpha right, then the resistance is going to be R alpha is 3 V DD by 4 into alpha times I DSAT

because I D SAT is proportional to W. if I increase my W I have more current linearly more current and therefore, 1 by alpha times the resistance, therefore this will be R by alpha. So, this is R by alpha, right.

So, what do I have on this NAND gate where I have assigned a unit width to both NMOS transistors, what is the net pull down resistance? It is 2 R, right. So, if you look out look at the pull down networks resistance, it is basically R and R. So, if I compare this to my reference inverters pull down resistance it is twice that right because this guy has pull down resistance of R. My design is to by design I want to make sure that the worst case pull down resistance on that pull down network is exactly equal to the pull down resistance of my reference CMOS inverter, ok.

Therefore I have to make this one as some alpha and we calculate 1 alpha is I am going to make it alpha alpha right, and the net resistance is now going to be what R by alpha R by alpha, then this will be 2 R by alpha, but by design I want this to be how much. R; therefore, alpha has to be 2 right and therefore, this will be 2, is that clear, yeah. So, the point is that ideally I would have like to size my transistors to the unit width because I want to say one area right. Why would I make it bigger? Right, I want that to be of course PMOS because mobility is half I have to double the width that we know.

Now if I make it unit width then the net pull down resistance for this NAND gate is R plus R which is 2 R. So, by design I want the worst case pull down resistance to be only as much as that offered by a reference CMOS inverter. The reference CMOS inverter is this guy. How much does he offers? He offers R I want the net worst case pull down resistance; remember, this is very important thing the worst case pull down resistance should be R.

It can be better than that there is no problem certain combinations it even be better, but the worst case should be R and therefore, I have no choice, but to double the size of my NAND to NMOS transistors in order to match this resistance, right. Therefore for NAND gate NMOS transistors both will have a size 2 for a NAND 2. What about the PMOS? What is the worst case pull up resistance?

Student: (Refer Time: 15:18).

No; so, what is the worst case constraint is it one of the transistor pulling up or both?

Student: (Refer Time: 15:24).

It is only one transistor pulling up because if both resistances are in parallel it will fall by 2 anyway, that is not the worst case. So, the worst case is one of the transistors pulling up and therefore, I can keep that size to 2 because that is the same as the reference CMOS inverter where I have one PMOS pulling up with a size 2. Here, again the worst case is just 1 PMOS and therefore, I can make the both PMOS size 2, ok.

So, we will stop here we will continue tomorrow with more examples of the NAND 3, NOR 3 and so on right where all of these become more clear.

Thank you.