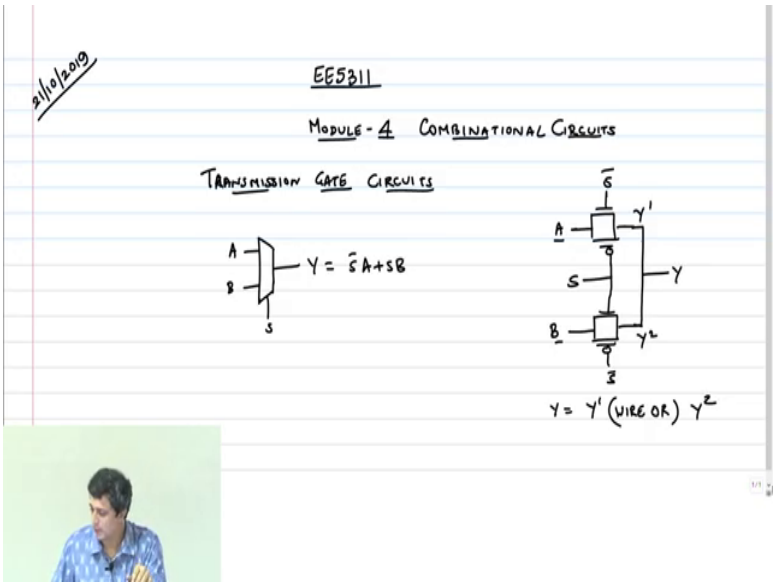
Digital IC Design
Prof. Janakiraman Viraraghavan
Department of Electrical Engineering
Indian Institute of Technology, Madras

Lecture - 52
Transmission Gate Logic

(Refer Slide Time: 00:15)



What I want to cover is the Transmission Gate Transmission Gate Circuits, ok. So, what is a typical 2 is to 1 multiplexer logic function if this is my select line A B Y equal to what?

Student: S bar A.

S bar A plus S B, right; now, if you want to implement this in static CMOS logic right you clearly have to you know implement this and of S bar into A plus S into B an inveterate all

that stuff, right. Similarly, if I now take this to a 4 is to 1 multiplexer it is going to become that much more complex because it will be S 1 bar S naught bar into A, S 1 bar S 1 into B and so on, and as I take the complexity higher and higher and higher the circuit it going to become really-really complex, right.

So, it turns out that when you want to implement something like a multiplexer then the best thing to use is a transmission gate, ok. So, the transmission gate circuit is like this right. So, we obviously, do not put only one NMOS or one PMOS, because NMOS can only passes 0, PMOS can only passes 1. So, it pass both logic levels you put them in parallel whichever is least resistance will pass it easily, right. So, this is the basic idea behind a multiplexer, transmission gate multiplexer, right. So, if you look at this, this is S and this is S bar.

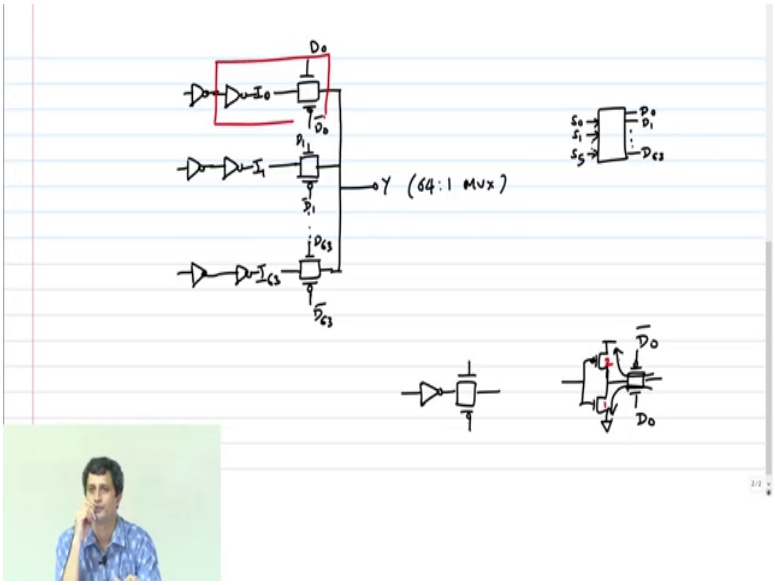
Student: (Refer Time: 02:15).

Right, the key thing that we are exploiting here right, I will leave it you to go and you know figure out how do you implement the static CMOS version of let us say 4 is to 1 mux. You will see how complex it is how many transistors you need, what the logical effort is and so on right, it is a big problem.

The main thing that we are exploiting here is if a transmission gate is off right then the output goes to a high impedance state right, I am driving 1 end with an input A or B, but if that path is off; that means, both the NMOS and PMOS are off then the output let me say Y 1 and Y 2, right. Y 1 and Y 2 will go into high impedance state in a mutually exclusive manner and therefore, I can do Y equal to Y 1 wire or of Y 2.

This is what is really saving me plenty of gates right because it goes into high impedance state I can do wire or connection and now I can do a wire or connection for any number of such transistors transmission gates.

(Refer Slide Time: 03:49)



For example: if I want to implement let say 64 is to 1 mux right what do I need to do I basically need to simply do this, this is I 1 right or I naught. We will fill the select lines later, I 1 all the way down to 64th I 63 right, and I can simply wire or all these connections and get my output Y which is a 64 is to 1 mux. Now, what are the select line for this; so, I need the decoder which is going to decode a 6 is to 2 power 6 you know circuit correct.

So, I will put a decoder which gives me the following select line S 0, S 1, S 5 and what I get here are my decoder signals D 0, D 1, D 63, but in order to turn on this fast transistor I need both 2 and complement outputs. I need the decoded output and the complement of the decoded output because for the PMOS I need that right therefore, I will basically make this as D 0, D 0 bar this is D 1, D 1 bar this is D 63 and D 63 bar, ok.

How do you implement a decoder by I when cannot be done you know just for example, if I want a 6 input NAND gate AND get I cannot do it, right because I effort for logical effort reasons. Therefore, you build this from smaller 3 is to 8 decoders, you build 3 is to 8 decoders then use them cascade them and then build your 6 is to 64 decoder, right.

So, like I told you this is the case where both 2 and complement outputs are needed from the decoder. So, typically here if you want to let us say do a decoding and then a level translation then CVSL is a natural choice because it also does level translation for you. Additionally, it also gives this true and compliment output that you need that your decoder that you are multiplexer needs eventually, ok.

The only problem with this style of circuit is if I ask you what is the drive strength of this multiplexer, what is your answer? What is the drive strength? If I have an inverter unit inverter then I say the drive strength is 1 1 x right, if I have a unit NAND gate drive strength is 1 again, you know you size it like that. Now, if I ask you what your drive strength of this transmission gate base multiplexer, what is it?

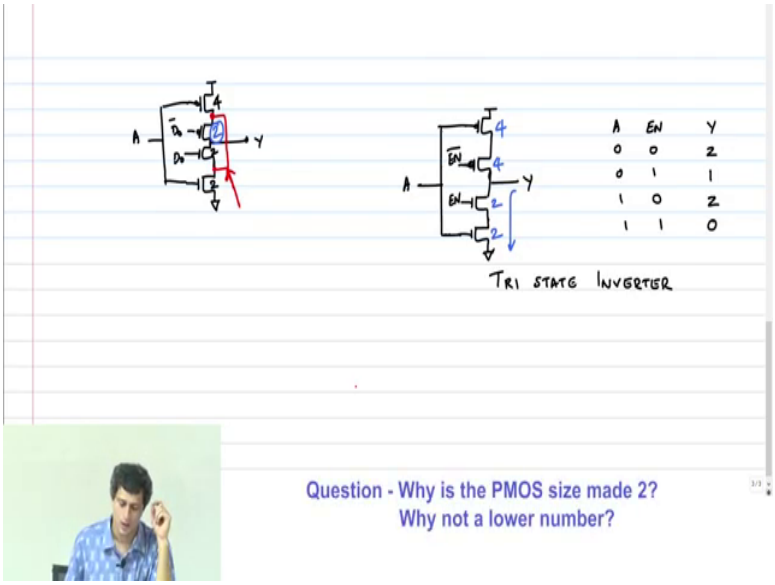
So, clearly that you cannot answer that question, because unless you tell me what is driving this gate I cannot tell you what the drive strength this gate is. For example, if I have an inverter here right or a buffer or whatever to be if you do not want inversions then put a buffer no problem. Then, I can now answer that question because if I look at this inverter right if look at this part of the circuit, it is going to be an inverter followed by a transmission gate.

So, if I open this up right and draw it a transistor level, this is my PMOS, NMOS, this is my transmission gate, this is my D naught, D naught bar. Now, you can answer that question because the PMOS is going to be pulled up like this, the NMOS is going to pulled up like this pull down like this. So, now, I can even answer the question what should the size of this PMOS transistor B right or the NMOS transistor B in the transmission gate. So, if this is let say 1 and 2, what should be transmission gates size B?

Student: (Refer Time: 08:59).

So, again what we have to do we have to this redraw this in a fashion that we already know, ok.

(Refer Slide Time: 09:11)



So, what I am going to do is I am going to right; this is my inverters input right if I go back here. I am just opening that up, opening this inverter up and fitting my transistors in there my transmission gate I am put in there, ok. So, that I get a stuck because that is what we are used to, right. So, this is my inverters is input a then this will be a right and this is going to be my output Y, this will be D naught, D naught bar. Am I done is this equivalent to that earlier circuit.

Student: (Refer Time: 10:19).

Why?

Student: (Refer Time: 10:24).

Yeah, I have to short the other 2 terminals that is why because it is it is a parallel thing right; so, I have to draw this also. Now, I will ask you the question what should the sizes B? 2 2 4; so, you would say this is 4 right, what about that 2 there, I mean should that PMOS be 4?

Student: (Refer Time: 10:55).

Technically, it is not necessary because the NMOS and PMOS are actually conducting together, even though the NMOS cannot pass a logic 1 completely for most of the time it is going to actually allow conduction until it go to V DD minus V T. The NMOS is also going to conduct along with the PMOS therefore; the two resistances are in parallel for most of the time.

And therefore, you can slightly given allowance and typically they make this also 2 slight deviation from what we are done, because these two NMOS and PMOS are in parallel for both logic states. This is very important because only do sequential circuits later we will use this, ok, clear.

Now, it turns out that you know you really do not need to make this parallel connection; I can still get a very valid gate that does exactly the same functionality even if I remove this parallel connection here, ok. So, if I re draw back circuit, it is going to look like this, one I need to make this separately this is my input enable bar. So, now what is the truth table of the circuit, I will do A enable and Y 0 0 0 1 1 0 1 1, what happens when enable is 0?

Student: (Refer Time: 13:19).

Output is what?

Student: (Refer Time: 13:23).

Yeah.

Student: 9 (Refer Time: 13:27).

Floating or high impendent state, when it is 0 1?

Student: (Refer Time: 13:34).

Yeah, when it is 0 1 when enable is basically 1 that gate basically is like an inverter, it just inverts input A, right; so, this is going to be 1 and this will be 0 and this will be Z. So, not surprisingly this is called a tri state inverter, because like I told you transmission get sometimes you know you need a gate driving it before where is that tri state inverter is a complete gate by itself. I can use it anywhere I can size it, I can be done with when is transmission gate have to be careful about where a place that gate and size it according to that circuit right, but both effectively perform very similar functionalities. And here now, the sizing should be what?

Student: (Refer Time: 14:40).

2 2 now it should be 4 4 I have no choice because I do not have that parallel conduction right when it is going to logic 1 only PMOS will conduct, when it is going to logic 0 only NMOS will conduct, this is very critical as you will see later, ok. So, you can use you know again the output Y therefore, of this circuit can be wired or to some other tri state inverter if you can ensure that they are mutually exclusive that is the advantage of this kind of a circuit ok, clear any questions here, yeah.

Student: (Refer Time: 15:23).

Here right yeah. So, if you look at this circuit even when I am passing logic 0, see in the tri state inverter when the output has to go low, the output only has to go like this only NMOS will conduct PMOS is turned off. Now, in my other case because I have this parallel connection this red parallel connection here, right. This red parallel connection here both the NMOS and PMOS of the transmission gate will conduct together though the NMOS will be like slower it will conduct only up to V DD minus V T for most of the time it is going to conduct, right.

So, in some sense they are both in parallel and therefore, you can reduce the size slightly. So, it is instead of 2 2 4 4 you are able to make it 2 2 2, 2 4 that is all, ok.

Student: (Refer Time: 16:26).

Why not? Yeah. I think yeah see this is not let me tell you this is not like some you know derivation where I can show that this is minimum, it is a it is in practice that you typically do this, that is all it is a reasonable thing. You have to figure out from simulation if maybe 1 2 is also ok, but you have to figure out from simulation that is all I am saying.