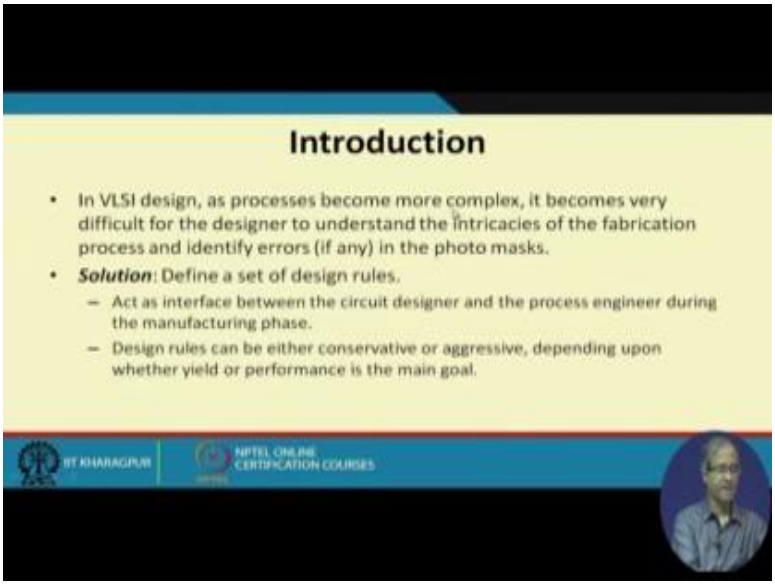
VLSI Physical Design
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 45
Design Rule Check

So, in this lecture we shall be looking at very importance step in you can say physical verification of the design, or some rules which the design I should follow in creating the layout is something called Design Rule Check.

(Refer Slide Time: 00:47)



Let us first try to understand what is a design rule in the context of VLSI layout design? The first sentence is important. You see as the processors are becoming complex, so on a layout we are having billions of this kind of rectangular shapes.

Now, how do we know that this billions of rectangular shapes the way they are laid out. So, it will result in the correct functioning of the circuits that we are trying to built, there can be so many issues there can be 2 wires can be too close together there can be short circuit or the width of the wire may not be wide enough due to some imperfection in fabrication that can be break or it has become too much narrow the resistance value as increased unnecessarily. So, there are lot of such issues which can take place during fabrication which might lead to the failure of the device or the circuits.

So, design rule says that while I can specify some rule of the thumb, like let say a wire which is being laid out on let say metal one I tell that the minimum length of the wire should be this and the minimum separation between 2 wires should be this, these will be my design rules. So, once the layout is created which means again a set of rectangular shapes on the different layers. So, using an automated tool I can verify whether this design rules are satisfied for all the rectangles, and the pairs of rectangles which can interact right. So, it is complex and time consuming process, but it is durable all right.

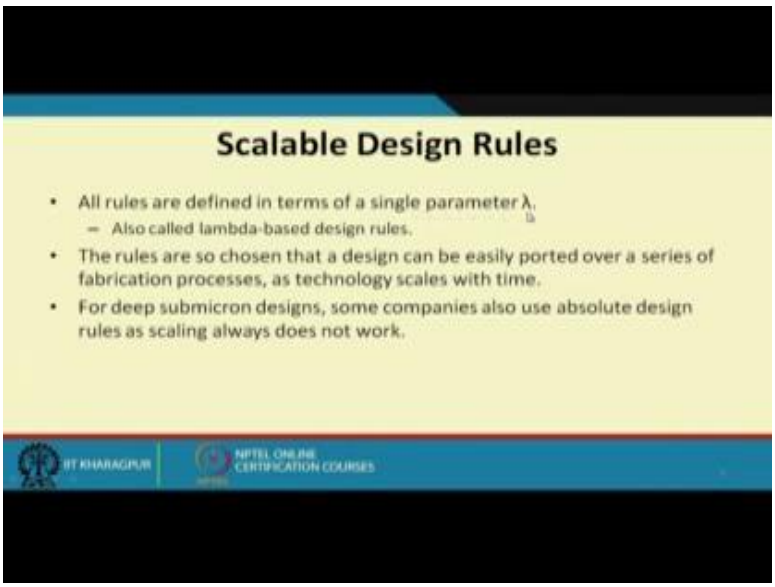
So, this is the basic purpose. So, it is difficult for the designer to understand the intricacies of the fabrication processes defaults that can occur there in. So, the solution is to divide define a set of design rules, these design rules they act as some kind of an interface between the designer and the person who is actually fabricating the process engineer. Design rules can be conservative or aggressive, conservative means I deliberately say that a wire length should be this which is more than what is required, because I am keeping the tolerance in mind even for fabrication process variations.

So, the wire width will be very little bit, but that little bit can be tolerated if I just initially to start to say that the wire width should be large enough this much. Similarly separation between the wires; if I specify they have to be separated sufficiently away, so even due to fabrication imperfection there will be no short circuit between them. So, this is the understanding, but the implication is that if you are conservative in this sense, the total means area will be increasing which means your performance due to delay will be increasing, but if you are aggressive you are not unnecessarily making the wires wider or separation larger, then you are talking about increasing the performance that possibly because of some failures, the yield at the percentage of the correct chips that are manufactured that might go down. So, this is the compromise.

(Refer Slide Time: 04:44)

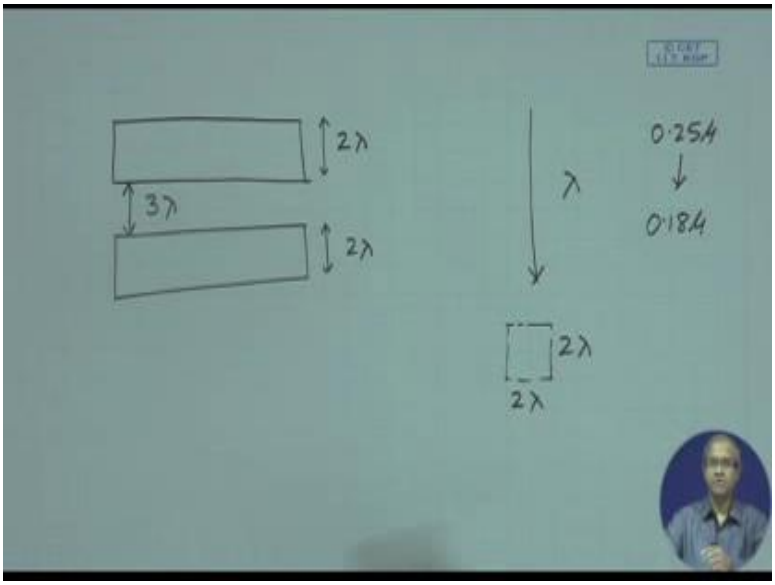
So, design rules actually will be looking at; this specify certain constraints which are geometric in nature with respect to the rectangles, we are calling it the layout here artwork. Now these rules they are typically minimum width and minimum separation or spacing rules, and they apply between wires or objects on the same layer or across different layers. Now if these rules are followed then it is ensured that the patterns on the process wafers will most likely preserve the topology and will lead to a correct design, so this becomes easier for the designer.

(Refer Slide Time: 05:35)



Now most of the design rules that have been used means in practice they are sometimes called scalable. Scalable means this rules are defined in terms of a parameter lambda.

(Refer Slide Time: 05:55)

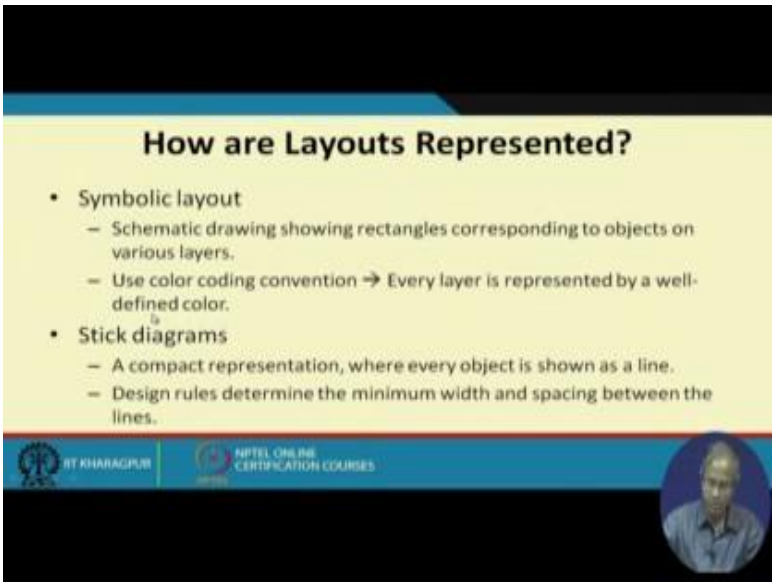


Like let us say. So, we specify that a wire let us say 2 wires running. So, the rule may say that the minimum width of the wire will be twice lambda, and the minimum separation between the wires will be 3 lambda but I do not say what is my value of lambda.

As my technology scales down devices becomes smaller and smaller, the value of lambda is also getting smaller and smaller. Roughly speaking lambda is indicative of the size of the smallest transistor, the channel of the smallest transistor is typically represented as 2 lambda by 2 lambda. So, half is smallest transistor size that can be treated as lambda. Now as technology scales down say from 0.25 micron technology, you had scaled down to 0.18 micron technology.

So, your design rules have not changed, but the value of lambda has changed. But of course, today in the area of the dips of micron technology, so only this scalable design rules will not work, there are some other design rules which are also augmented, which are based on some absolute values not based on lambda any more. So, results become more complex. So, this is what I mentioned the rules are ported or can be ported over a series of processes technology scales with time, and for deep submicron design we may be force to use some absolute design rules, because scaling for all the layers does not work equally well there.

(Refer Slide Time: 07:48)

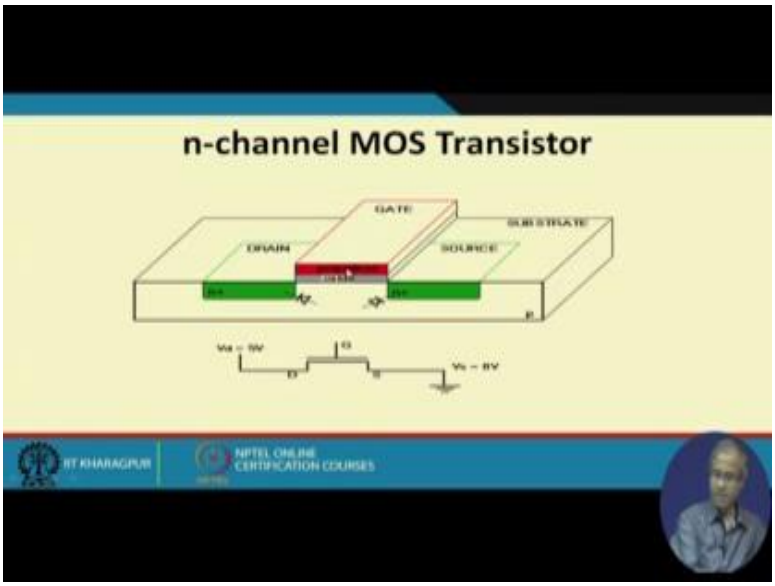


Now, how are layouts represented? We have already seen this during our earlier lectures, so we can use a schematic layout, symbolic your schematic layout (Refer Time: 08:06) thing where the layout is represented by rectangle corresponding to the different objects and we use some color coding convention. Every layer is represented by a different color like you recall the diagram we had seen earlier again, here a CMOS inverter we had represented in schematic form like this where do we use some color coding. Blue means metal, red means polysilicon, grey means diffusion, mean n diffusion and on this side brown is the convention for p diffusion and contact cuts black.

We have also seen that symbolic layout can be represented in a more compact to the stick diagram, by every object is shown as a line; like you recall we had seen that this same layout can be represented in a compact way by an a stick diagram like this. Now suppose we have access to the design rules, we already mentioned that a metal line must be minimum 3 lambda wide. So, here even if I shown it as a single line I know that this rectangle width will be 3 lambda. So, if I know that the separation between this metal and polysilicon line will be 3 lambda, then I already know this separation will be 3 lambda.

So, from the design rule I can directly convert my stick diagram into my layout right this is the advantage we gain.

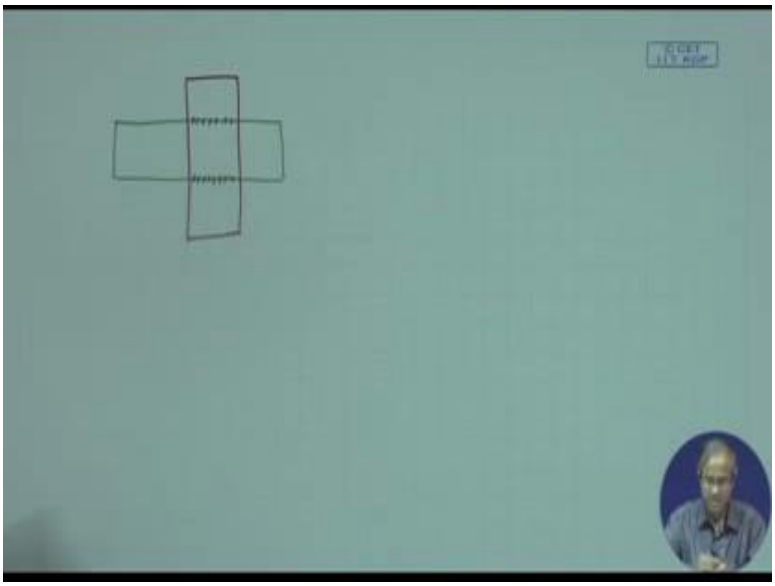
(Refer Slide Time: 09:45)



So, let us very quickly look at the MOS transistor fabrication basic. So, you know that n-channel transistor is created like this where so on a substrate, p type substrate there are 2 diffusion regions highly docked n regions I created. So, one is the source, one is the drain and in between there is an oxide insulation layer on top of which a polysilicon layer is created which forms the gate.

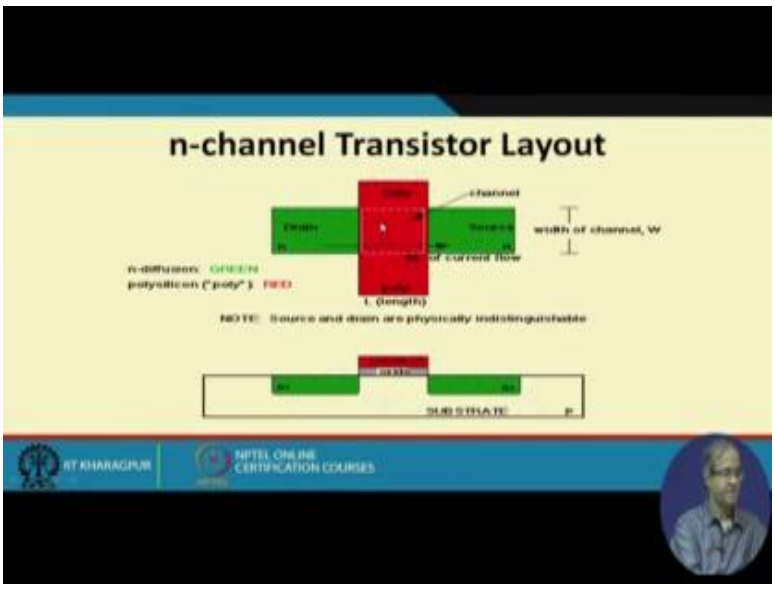
So, by applying a suitable voltage on the gates, so electrons can be drawn into the channel and the channel can conduct or not conduct, equivalence circuits looks like this. Now you see for a transistor so it is actually like this, there is diffusion on one side, diffusion on the other side nothing in between and top there is a solid polysilicon block.

(Refer Slide Time: 10:49)



But in a layout when we show we do not show the disconnection in between, we show the diffusion as a solid rectangle and we show the polysilicon also as a solid rectangle every bit. But it is understood that in the region which is overlapping, the diffusion green region is actually not there, but this is just a matter of convention.

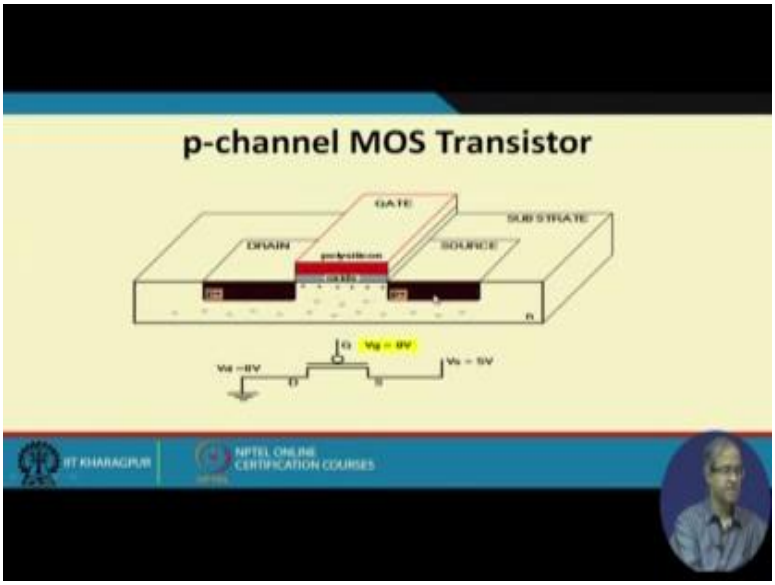
(Refer Slide Time: 11:22)



So, in our symbolic layout or schematic layout we show it like this a continuous rectangle, this is what I present. So, the transistor like here is represented on the

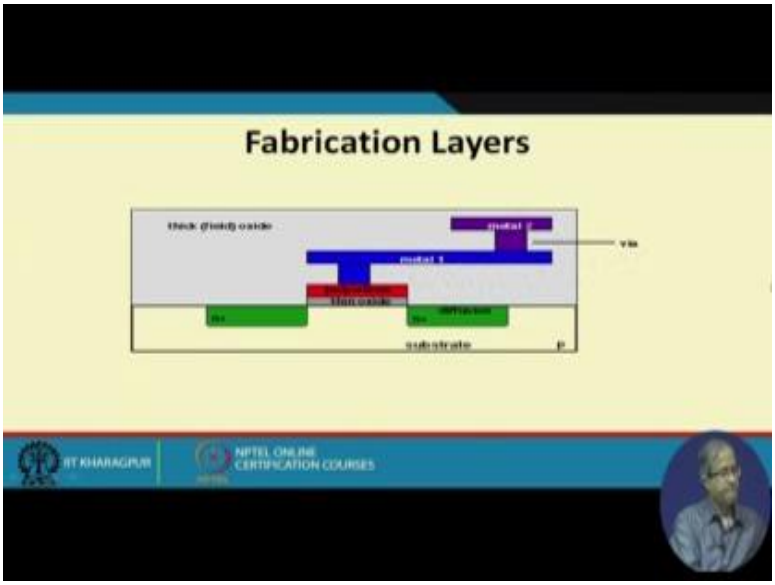
schematic like this the solid this is the top view; diffusion on the lower layer and top of which the polysilicon layer in red, which is the gate.

(Refer Slide Time: 11:46)



Similarly, p type MOS transistor is quite similar, just this is created on top of an n type substrate or a well, there a p type diffusion regions and similarly oxide. So, the color convention for the diffusion region here is brown.

(Refer Slide Time: 12:08)



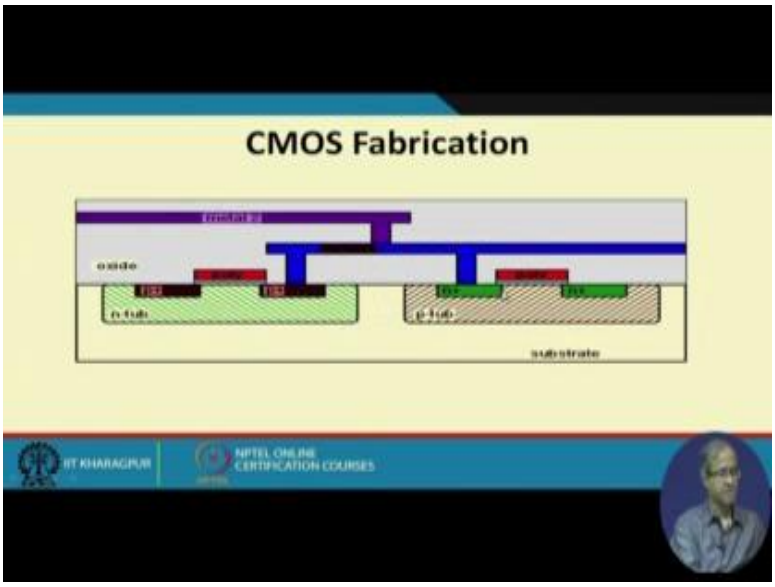
And just above diffusion and polysilicon layer there can be several metal layers, will be metal 1 metal 2, 3, 4, 5 and the remaining space there is insulating oxide you call it thick

oxide and as then when required there can be connections across layers like this connection shows between metal 1 polysilicon, this shows between metal 1 and metal 2, these are wire connection or contact connections.

(Refer Slide Time: 12:42)

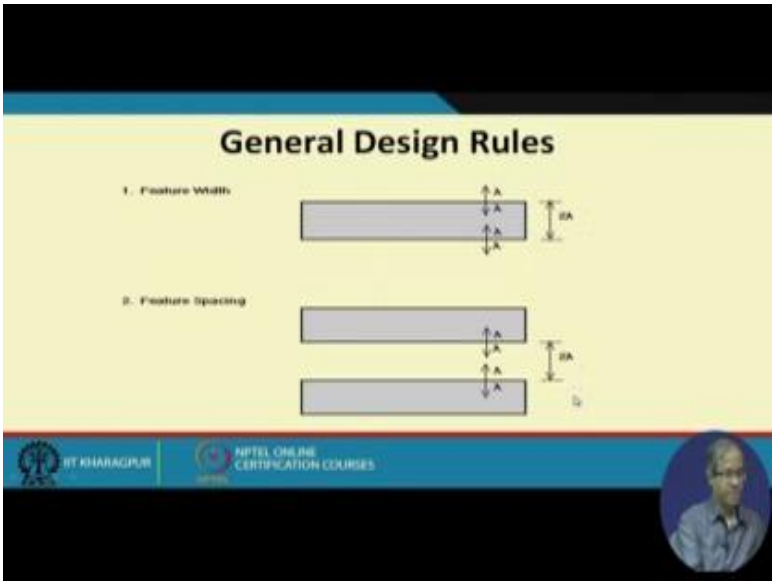
These are the conventions that I had talked about. So, a polysilicon wire is represented in red diffusion for N type diffusion is green P type diffusion in brown, metal 1 blue, metal 2 purple and the Via's the connections contact represent black.

(Refer Slide Time: 13:06)



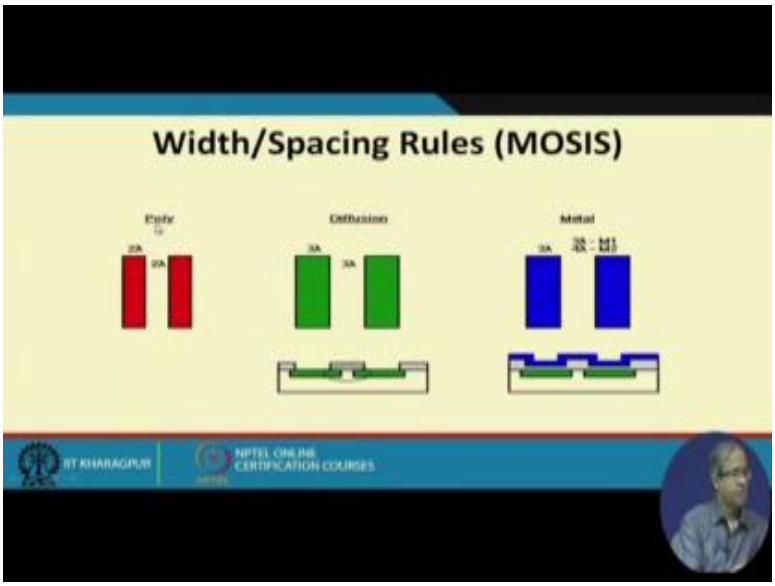
So, this is just another example which shows 2 transistors one n type, one p type with the source of one transistor connected to the drain of the other transistor. So, maybe we are trying to build an inverter. So, this is this central point from where we will be taking the output you of course, the input connections are not shown, but these are incomplete inverter right.

(Refer Slide Time: 13:36)



Now, the design rules as I have said they will look like this. First kind of design rule is specify feature width given a rectangle shape, so what should be the minimum width? See these are all minimum, so there is no harm if I use 3 lambda in place of 2 lambda. Features spacing between 2 such shapes rectangles. So, what should be the minimum separation? So, here it is shown as 2 lambda this is example. Now depending on the layer this values will change.

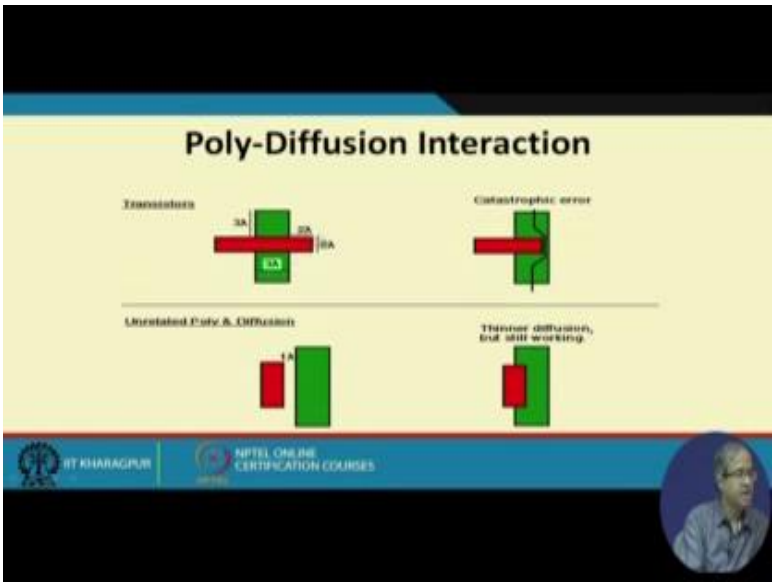
(Refer Slide Time: 14:19)



So, some actually practical design rules are shown here, this is based on a standard design rule called MOSIS.

So, this rules specify like this, polysilicon where gates are formed the minimum width has to be 2 lambda, separation is also 2 lambda, diffusion which is down below, diffusion the minimum width is 3 lambda, separation between 2 parallel diffusion must also be 3 lambda. Metal 1 as you move up 3 lambda, but the separation is 3lambda for metal one, but as you move up to metal 2 separation has to be more 4 lambda. So, as you move up the layer, the separation also starts to increase this also you should remember.

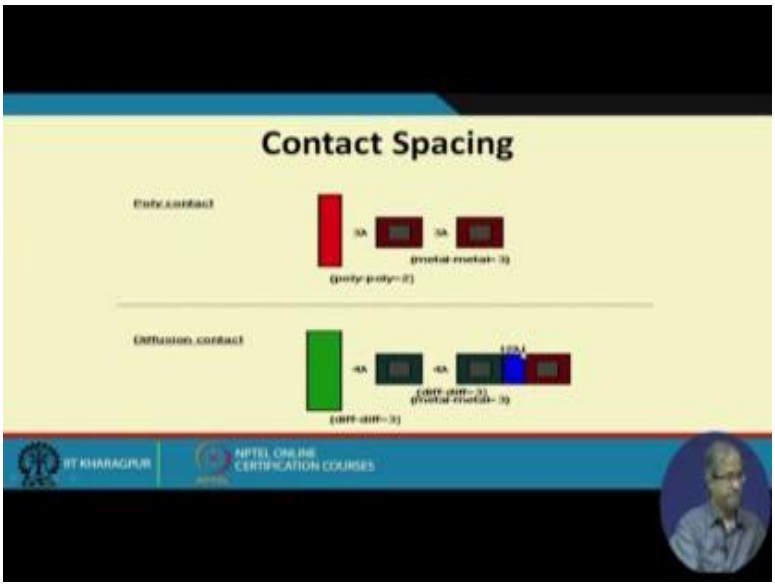
(Refer Slide Time: 15:14)



Well here the transistors are formed. So, diffusion region and a polysilicon region is intersecting now you see. So, when a diffusion and polysilicon is intersecting we have mentioned a few things, but diffusion can be 3 lambda y in width, and a polysilicon can be 2 lambda in width. So, this is the minimum; 3 lambda here 2 lambda here, and regarding overlap on this side minimum 2 lambda should be there, on this side minimum 3 lambda should be there. So, if you do not keep this extra separation means extra this polysilicon region here then because of some fabrication error you may land up in a situation like this, where the polysilicon is not covering the whole channel and there will be a current flowing path irrespective of the voltage you are applying to the gate, right.

Similarly, when there are 2 polysilicon and diffusion layers these are not related and the 2 layers, they must be a 1 lambda separation between them. So, it is not desirable to have an overlapping like this, because it may accidently form a transistor; so there as to be a separation.

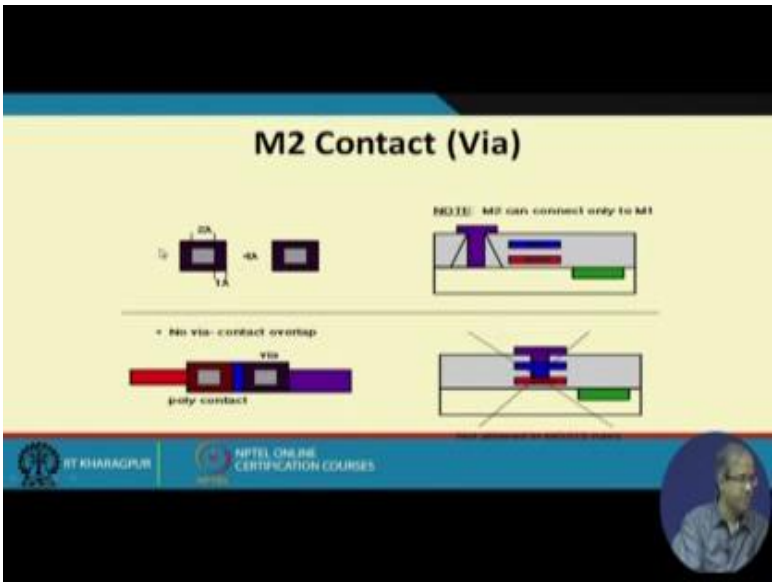
(Refer Slide Time: 16:38)



Similarly, for contact spacing; so spacing between contacts see there should be 3 lambda spacing between 2 contact connections, and if there is a polysilicon line running there should be again be a 3 lambda separation between this polysilicon and this contacts this is for poly contact polysilicon contact polysilicon layer to metal.

But for diffusion contact diffusion to metal the separation should be 4 lambda; from one contact to the other 4 lambda again from diffusion region to a contact 4 lambda. So, if there is a polysilicon contact and a diffusion contact side by side, then there has to be a region of overlap which must be minimum 2 lambda.

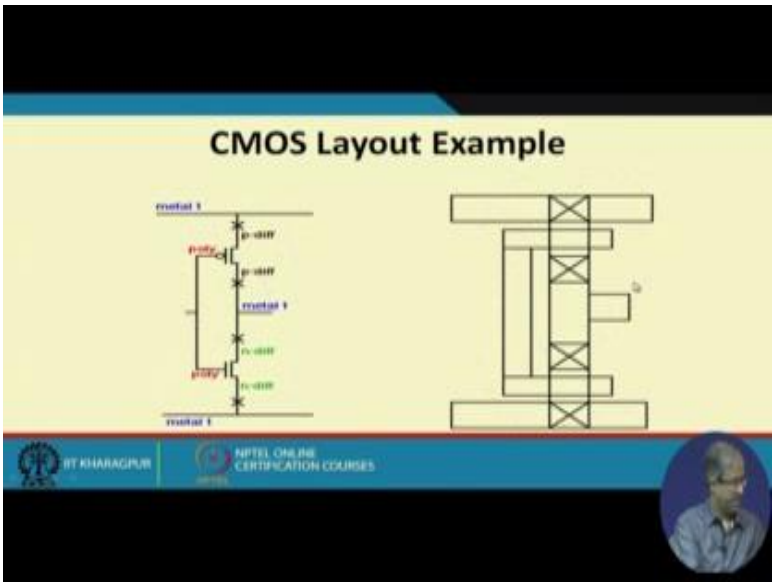
(Refer Slide Time: 17:31)



Now, regarding the via connection. So, if you want to connect M 2 to M 1, then there for 2 such contacts separation between them should the 4 lambda and the size of the contact the rectangle internally should be 2 lambda by 2 lambda, and separation on the 4 side should be 1 lambda each.

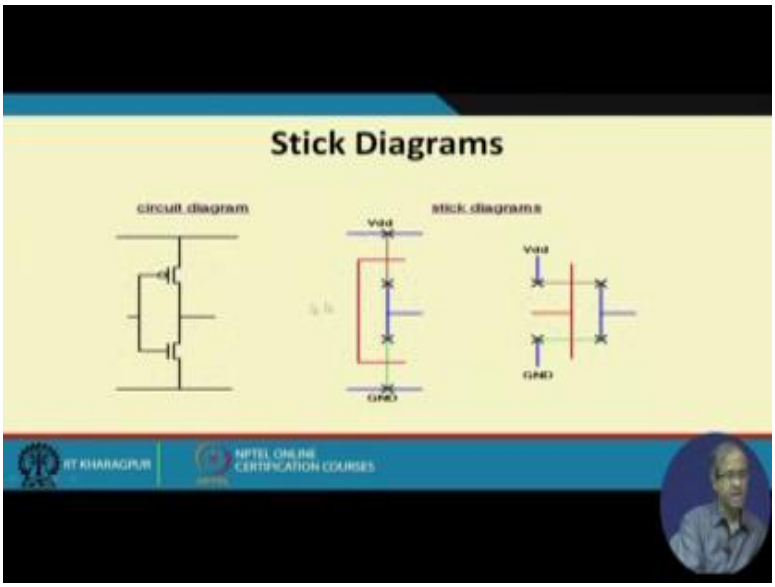
Now, this is something which is not allowed; that means you cannot directly connect a polysilicon and a metal layer with a via connection, via is only between 2 metals: metal 1 to metal 2. So, for connecting our polysilicon with a metal you have to use a contact like this right 3 lambda, fine.

(Refer Slide Time: 18:21)



So, this is what we have already seen earlier right. So, we can have a CMOS transistor, COMS inverter using an nMOS and pMOS transistor, you can have a schematic layout has we have seen already and schematic layout can also be represented by stick diagram.

(Refer Slide Time: 18:41)

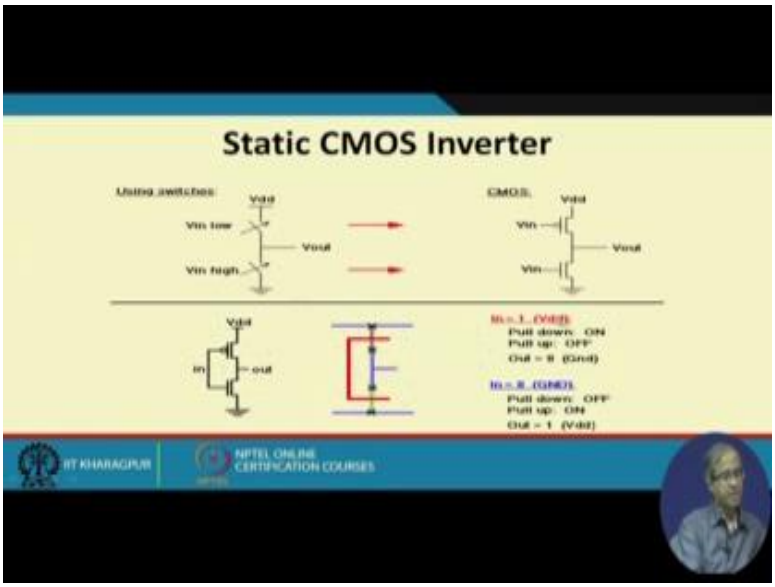


Now, as you can see that for this same inverter there can be multiple ways to layout, like here I have shown 2 possible stick diagrams; like here say instead of directly connecting as I was showing earlier, this diffusion p diffusion and n diffusion with a contact, so which is difficult? So, here I am showing a metal line in between. So, actually you are

doing a diffusion to metal contact, diffusion to metal contact and taking the output on metal. Now instead of doing it like this you can also doing it like this, the 2 layers instead of vertically diffusion and the poly this, this the p diffusion and n diffusion you can put them horizontally, and instead of 2 segments are polysilicon you can put a single vertical segment of polysilicon that will be crossing both of them.

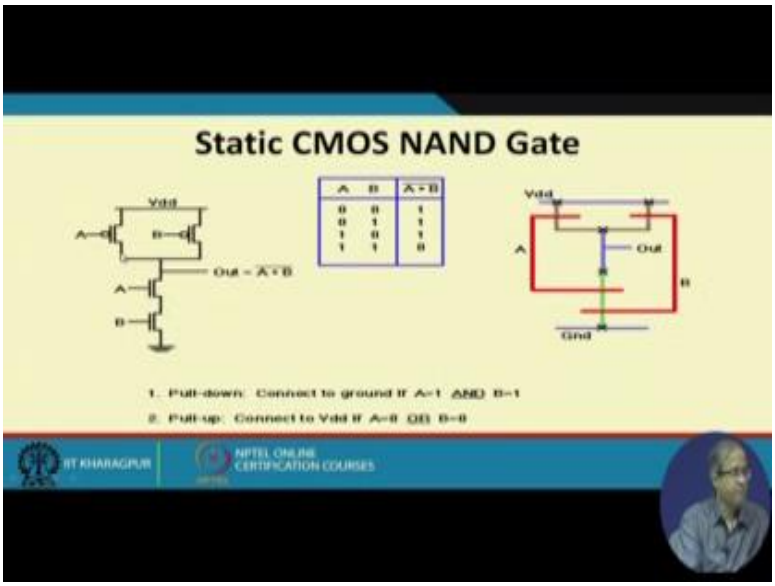
So, there are multiple ways of creating the layout. So, there are issues your layout has to be compact smaller in area, if this second one is better than the first one, because here you are here height of the layout was becoming larger right. So, let us look at.

(Refer Slide Time: 20:02)



This is already we have mentioned, this is your inverter this is your stick diagram depending on whether you applying 1 or 0. So, one of the transistor will be conducting and the output will be just the compliment of the input.

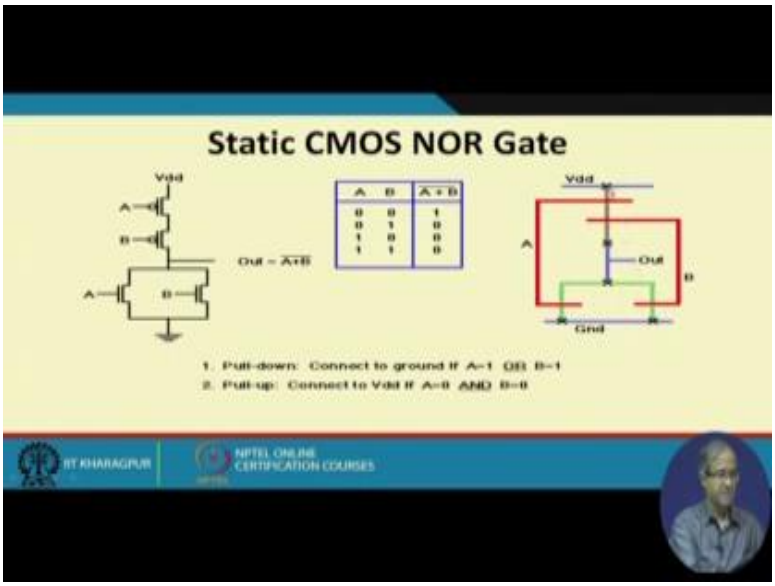
(Refer Slide Time: 20:23)



Let us look at some more examples. So, here I am showing CMOS nand gate. So, you see the CMOS level schematic is like this there are 2 p type transistors in the pull up, and 2 I means n type transistor in the pull down. Now again this is a not a unique thing, so I have shown one possible stick diagram to realize this. So, for this A and B transistors there are 2 segments of p diffusion you can see; this A is crossing here creating a transistor, B is crossing this one creating a transistor and these are getting connected this point the 2 are getting connected and then there is a metal connection which connects it to the pull down network there is a metal connection.

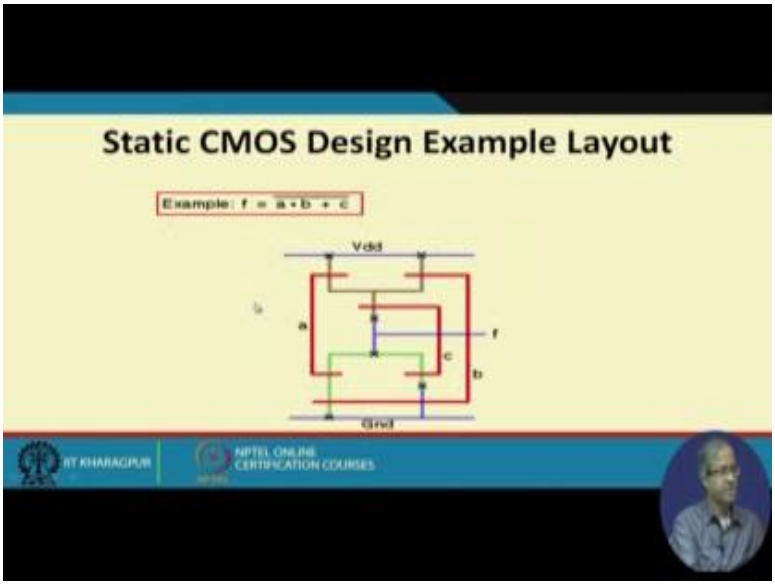
So, pull down networks has this n diffusion vertically, this A is crossing creating the first transistor here, and B is crossing creating the second transistor here. So, you can see this is just like doing an artwork. This is a way; so there is so many ways you can create this structure. So, you will have to correspond to this netlist, and will have to generate the connection on the different layers, so that you can create the transistors you can create connection to V DD and ground and also connection to the output and the inputs.

(Refer Slide Time: 22:05)



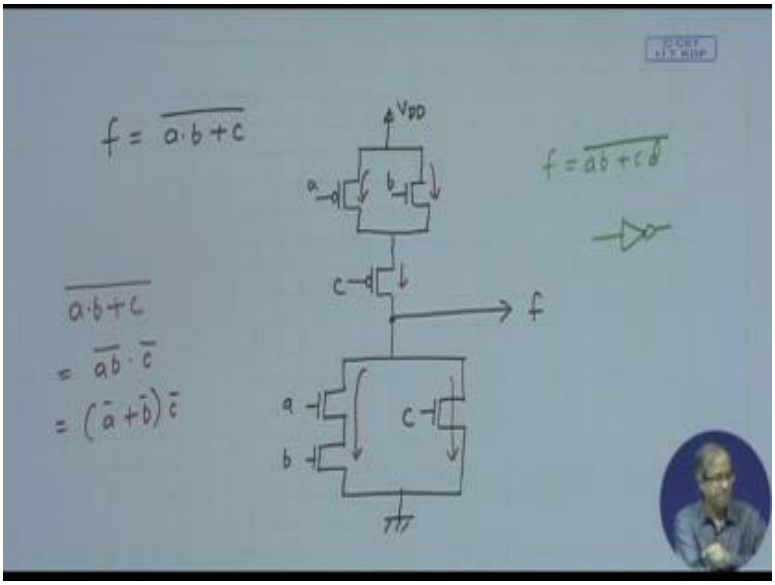
Let us take another example - NOR gate; so, nor gate looks like this. So, you see nand and nor the layouts are just reserve, the p type which was above so it will like a mirror image about a horizontal axis. So, layout becomes like this. So, now, the n type transistors are coming in parallel, the p type transistors are coming in series. So, now, the issue is that if you have a scenario, where there are more complex gates you are trying to implement.

(Refer Slide Time: 22:48)



Let us take one example here, suppose we are implementing a function a b or c bar like let us just look at it once.

(Refer Slide Time: 22:57)



So, we are implementing a function a b or c bar. Now exactly what will be this CMOS level circuit diagram for this, there will be a pull up and a pull down network, let me show you the pull down network first. Pull down networks will be having transistors like this, where a and b will be coming in series, c will be coming in parallel; but in the pull up networks it will be just the dual of this, dual means this will be your c it will coming in series, and then a b will coming in parallel something like this.

So, you take the output from here f. So, this function they wait works is like this if this a b or c this condition is true; that means, either both a and b are 1 or c is 1, then the pull down networks will be conducting if both a and b are 1, this path will be on if c is 1 this path will be on so f will be connected to ground. Let a b or c bar means if you apply De Morgan law, it will be a b bar c bar which is a bar or b bar c bar. So, the pull up networks implements this. So, if you can say a and b anyone of them is 0, which means either this is conducting or this is conducting and c is 0 which means this is also conducting then V DD will be connected to f. So, output will be 1, right.

So, this is a CMOS level schematic which you can generate very easily from a given function, but as you can see that the layout may not be that obvious. So, generating a layout for a complex function without the lines crossing each other is a challenging task.

So, if you analyze this it will look to be correct, but given in a arbitrary function how to get the best layout that is not a very easy task. So, you can see here there are 2 transistors in parallel coming in the pull up, then one transistors series here; and in the pull down a and b are in series and c in parallel.

So, you can have contacts in different see here there is a interesting thing here, you require a contact here and on metal you are bringing here why? Because already a polysilicon wire was running like this, and if you directly brought this diffusion green wire down here then another transistor would have formed here there is green and this red would be crossing. So, you have to be careful about these things. So, for a CMOS design, the generation of the CMOS level netlist like in this example which I had shown this is fairly straightforward. Given any negative function CMOS cannot implement a positive function like if you have a function like f equal to a b or c d, so a single CMOS get cannot implement this.

So, you will have to implement first the knot of this, then again an inverter after that; but translating from this into a layout is not easy, but the intermediate step may be easier for you this topology you directly map into a stick diagram, and from the stick diagram you try to generate the layout that is typically what you see this as a correspondence with the topology of the schematic which I have drawn this one right. So, they approximately represent a similar structure. So, this is what we wanted to know and design rules help in this ones you have a layout, you can automatically convert the layout or the stick diagram layout into the actual layout where the width and separation of the wires are defined.

So, design rule check helps you in 2 things: one in automated generation of the layout and after doing a lot of transformations like; you will see later we will talk about layout compaction. So, after doing some compaction well there might be some problems arising somewhere in the chip and because of that single problem the entire chip might fail. So, at the end you can also carry out a design rule check to find out if any such design rule variations, violations have actually taken place, right.

So, with this we come to the end of this lecture now in the next lecture as I have said we shall be looking into different ways of reducing the area of the layout compaction.

Thank you.