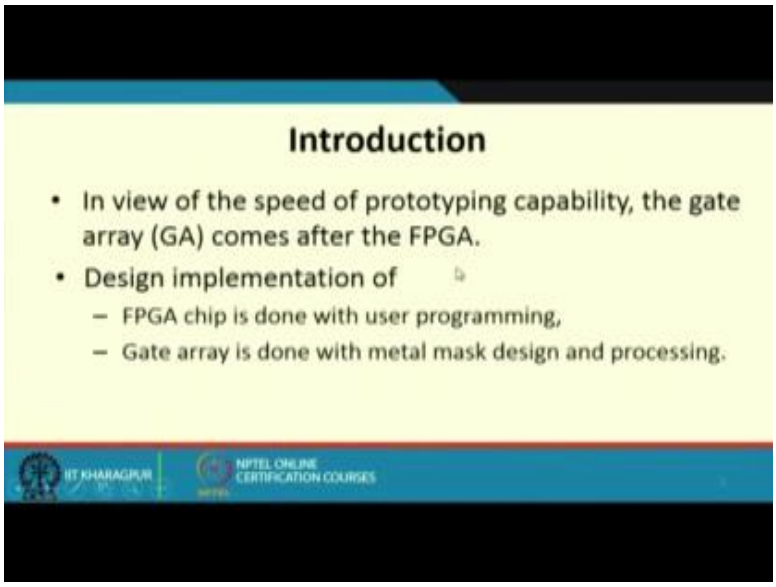
VLSI Physical Design
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 04
VLSI Design Styles (Part 2)

So, continue with the design styles. And, in the last lecture we had look at FPGAs; so how they work, what it is. And, in this lecture in the part two of this VLSI design style, this sub topic. So, we shall be looking at some other design styles. To start with we shall be looking at gate arrays.

(Refer Slide Time: 00:54)



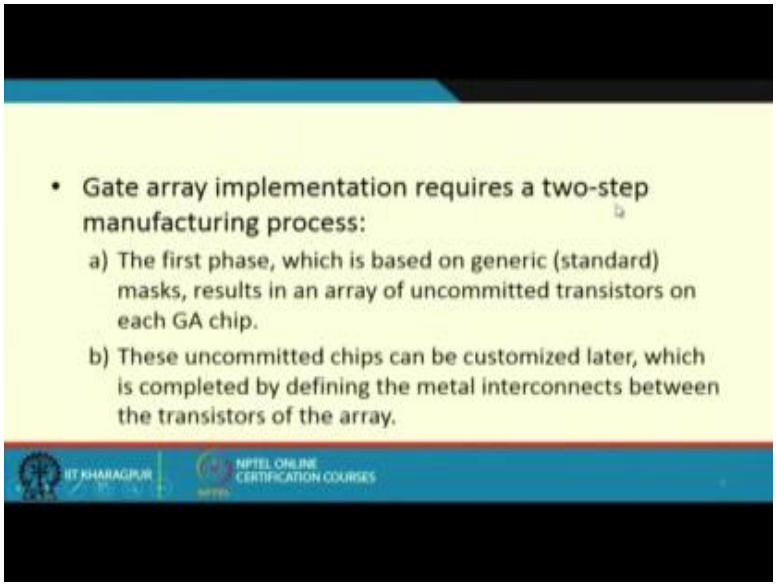
Gate arrays are slightly different from FPGAs. In this sense that it comes a little after FPGA. You see let us come back to FPGA and see what it was. FPGA was some kind of a design facility that is provided to us. Using which you can create a design very fast. But, the down side was that all logic we recall was implemented using LUTs, which are nothing but static memories in between. The LUTs are implemented using SRAMs. So, naturally they cannot be as fast as some logic implemented using logic gates. So, FPGAs will be slower in terms of the speed of operation.

Now, here when you are talking about this second technology called gate arrays or GA, gate array will be little less flexible then FPGA. But, its speed will be a little higher. So, these are all design performance tradeoffs.

So, in that sense we say that in terms of the speed of prototyping, gate array comes after the FPGA in term, in the sense that it is slightly slower. The main difference is that to implement a design on the hardware. For FPGA, you can do it in your lab. You can do it with user programming. So, if you have the CAD tool available with you, you can do it yourself.

But, gate arrays cannot be done in that way. You have to send your information or a request to the fabrication facility to get it done. But, you may ask the question that will if I have to send it with the fabrication facility, then what is the advantage of having this gate array? Why do not we design and develop a more conventional chip and send it to the fabrication facility. Now, we shall see the basic principle of gate array; the way it works the way it is created, the total cost becomes much less as compared to a complete designing an IC from scratch kind of a philosophy. Let us see how it happens.

(Refer Slide Time: 03:31)



Gate array requires a two-step manufacturing process. This is very important. The two steps are like the first step is design independent and the second step is more like customization. This is design dependent. Like, I will come to this. So, what I mean to say is that suppose there are five set people organizations or customers, there are five customers, who want to manufacture the designs using GAs or gate arrays.

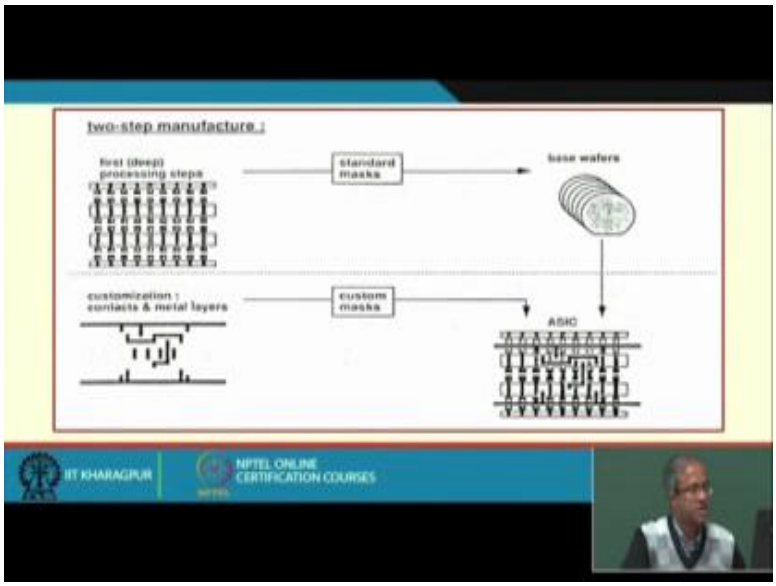
So manufacturing process, as I said comprises of two steps. The first one is independent of the design. So, the cost of the first step can be shared by these five customers. So, the

per customer's cost become much less and, in fact this first step of fabrication is the more complex step or the more expensive step, which is shared in terms of cost. But, once this first step is done, in the second step we take this specific request from the customers and we customize our IC or chip according to the request by the customers. So, the second step will be on a per customer basis that will be specific to the customers.

So, as you can see your cost component gets divided or reduced, again coming back to the slide. The first phase is based on something called generic masks. The first phase is called generic masks means we use a set of mask doing fabrication, using which you create some transistors on the chip; large number, millions of transistors. But, the transistors are not interconnected. You are not interconnecting them. Just fabricating the transistors in an isolation, in an isolatory.

In the second part where you have a design to be implemented, you actually complete the interconnections to create your designs. This is the basic idea. So, the first step you create a large number of transistors, which are not connected; in the second step, you to complete the connection, interconnection of the transistors depending on the designs that have been provided.

(Refer Slide Time: 06:22)

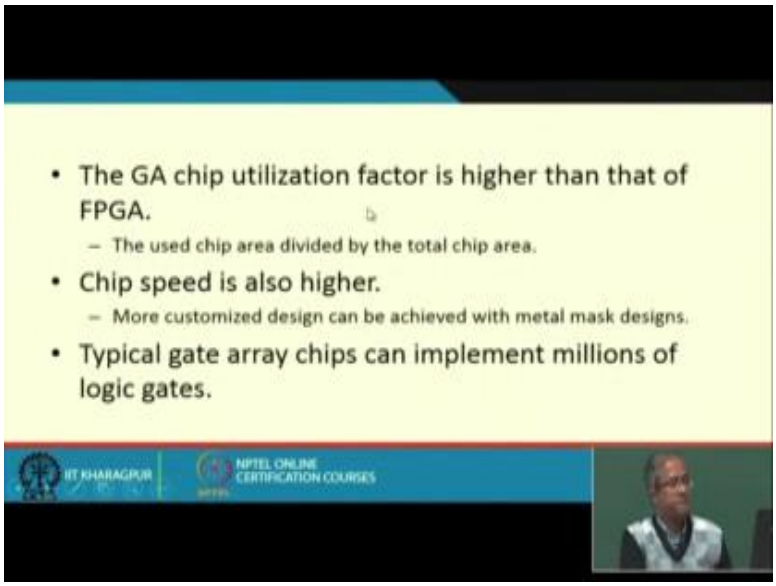


So, diagrammatically I am just briefly showing like this. This is the first step where you create a large number of transistors. Now, this transistor when you are creating, these are sometimes called deep processing steps because you have to create a number of layers

starting from the oxide layer, diffusion, polysilicon, oxide, then again, and just for the gate you need an isolation. So, number of layers of fabrication are required, which increases the cost of this step.

So, for this step you use a set up standard masks, which will be same for all the designs. And, you create the wafers, a large number of wafers which are common. They can be used by any customer. Now, in the second step you take this specific request from the customers, you create the custom masks and you only create the metallizations. You only fabricate the metal layers on top of the transistors that you already created in order to complete the interconnections. So, this is how this two-step manufacturing works.

(Refer Slide Time: 07:46)



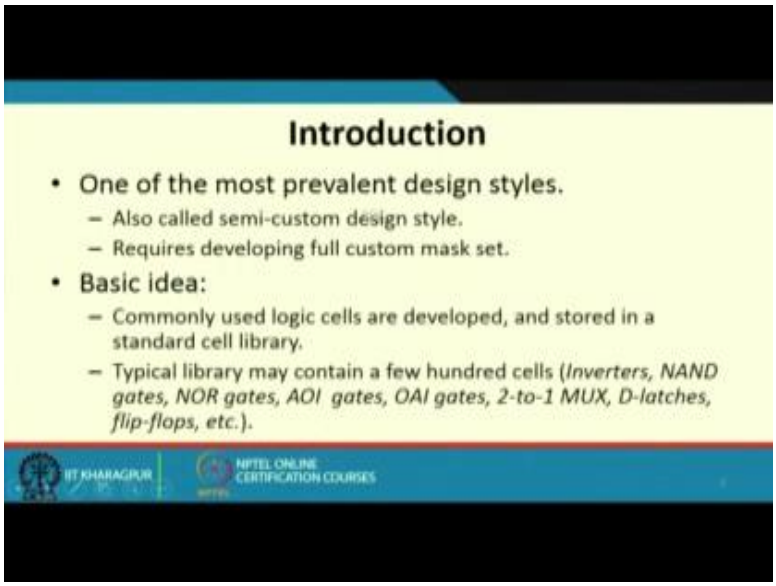
So obviously for a gate array, the chip utilization is higher because in GA you have transistors. And the transistors, you can utilize in any way you can. But, in FPGA you recall, inside this CLBs you have lot of unnecessary things. You always do not need. There are so many multiplexers, flip flops. You often do not need means all of them, many of them. But, in GA it is not like that. You can use any transistor when you need. And, obviously chip speed is higher because you are not using any static RAM anywhere. It is only the transistors which are connected. And, typical gate array chips can be pretty large in size, fine.

Now, let us come to design style, which is perhaps the most widely used today when we talk about creating the ICs. So, this standard cell design is one of the most prevalent

design style. In one sense, you see to tackle and handle the modern day complexity, no one designs the chips from scratch. There is a concept of design reuse. So, now people talk about creating something called technology library, where some of the small standard designs I have already created in a very efficient way. I have created a very compact layout for them and I have stored them in the library.

Let us say for example, a three input NAND gate. So, in my design whenever I need a three input NAND, I simply pick it up from the library, I put it in my layout. I need for input NAND, I again pick it up from the library and I put it in my design. This is the concept behind this so- called semi-custom design style.

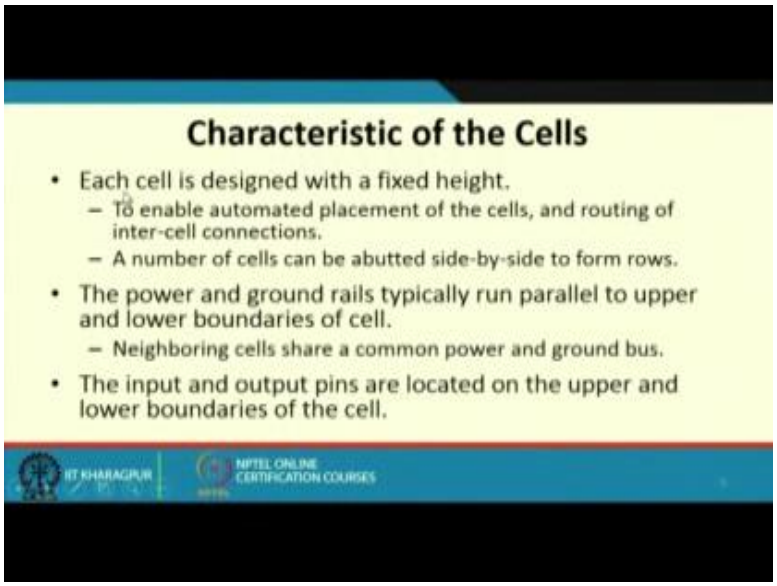
(Refer Slide Time: 09:53)



So, standard cell or semi-custom design style. This is a design style where all the masks have to be created because the cost of fabrication has to be borne by you only. So, so means you are a designer, you want to fabricate, so the entire cost of fabrication have to be borne by you.

So basic idea, as I have mentioned you develop the commonly used logic cells and stored them in a so-called cell library. A typical cell library can contain a few hundred cells. Some of them are shown. Some that the basic gates, multiplexers, flip flops say full adder; this kind of simple blocks are stored in the cell library, in terms of you can say highly optimized layouts. There are some constraints; we shall be talking about. So, this is the basic idea.

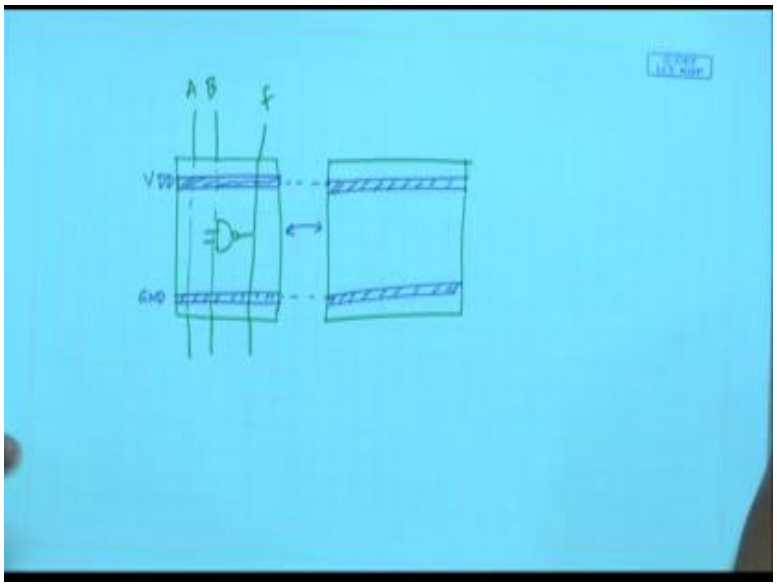
(Refer Slide Time: 11:01)



But, as I said there are some constraints regarding the cells. Each cell must have the same height. All the cells; their heights must be same, fixed height. Why? If this cells have fixed heights, you can put them side by side and joint them together. This is the advantage.

The cells have the same height with your VDD and ground lines running in the same, you can say, same horizontal configure. So that, if you put them side by side the VDD and ground lines will also join. This is one advantage or one feature. So, this allows automated placement of the cells along rows and interconnection. So, as I said number of cells can be put side by side abutted to form rows. The power and ground rails typically run parallel to upper and lower boundaries of the cell. Neighboring cells share common power and ground bus. And, input and output pins are located on the upper and lower boundaries of the cell.

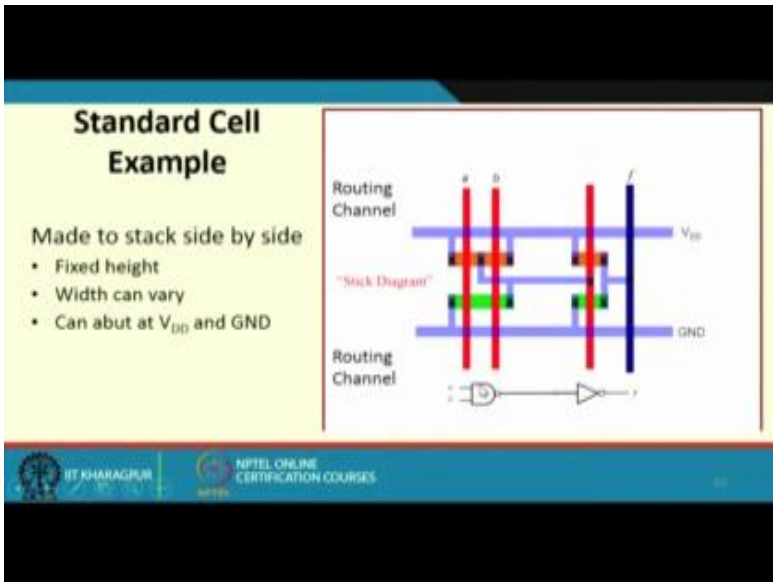
(Refer Slide Time: 12:15)



So, what I mean to say is that let me just show you once. Suppose this is one standard cell, I am just showing the layout as a box. Let us say this corresponds to a two input NAND. So, there will be some vertical lines that will represent the two inputs A and B. And, there will be another vertical line that will represent the output. Let us say f.

Not only that, as I said there will be a one fixed track for the power supply VDD and another fixed track for ground. So, if you look at some other standard cell, maybe some other functionality. So, this will also look very similar. And, the VDD and ground lines will be running similarly in the exact same horizontal positions, so that if you put them side by side, bring them closer together, this VDD and ground lines will touch each other. This is the main advantage. So, you can actually put all these standard cells side by side, if you require in rows, so that VDD and ground lines will automatically touch each other. So, they will get the power connections from there.

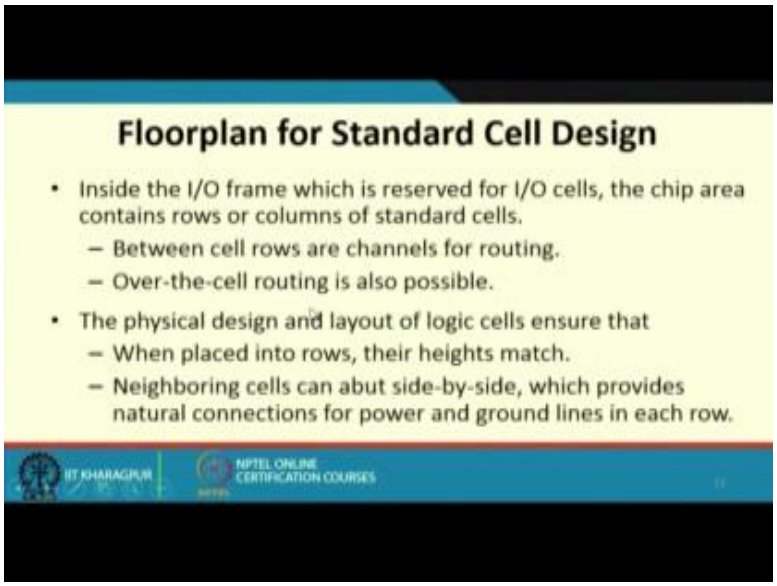
(Refer Slide Time: 13:54)



So, here some example is shown. This is the layout diagram of a two input NAND followed by a NOT, inverter. This is the CMOS design of a two input NAND. This red lines are the polysilicon lines, which feed the inputs a and b. And, this red line here is the input of this NOT gate, which is taken from the output of this first NAND gate. And, the output of this is available on this vertical line f.

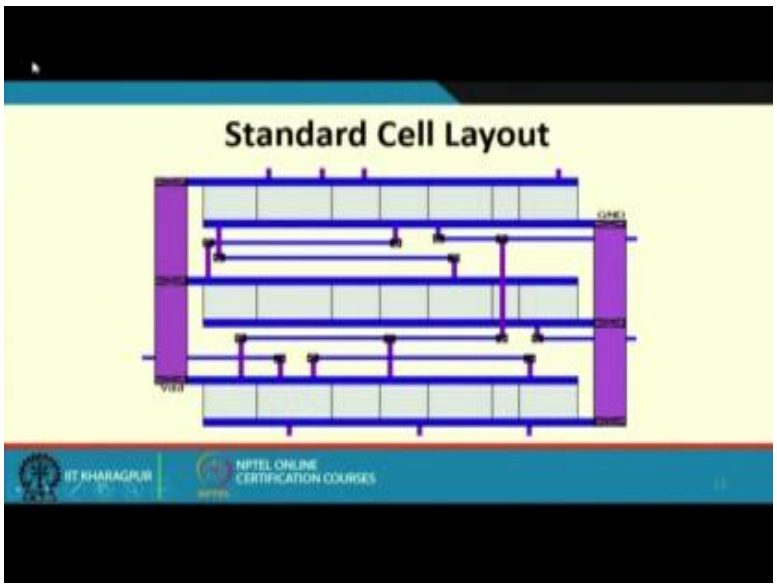
So, this is the example of a cell where a NAND gate and an inverter are connected one after to the other. So, this standard cells as a set can be of fixed height, but depending on their complexity can be of varying widths. And, they connected to VDD and ground. They connect together.

(Refer Slide Time: 14:56)



Now, when you talk about the floor planning for standard cell designs, so just inside the IO frame, what I mean to say is that you have the total chip area with you. So, this standard cell can be connected side by side. So, you can easily connect them in a number of rows. There will be number of rows in which the standard cells can be placed and there will be some space in between which can be used for further interconnections.

(Refer Slide Time: 15:31)

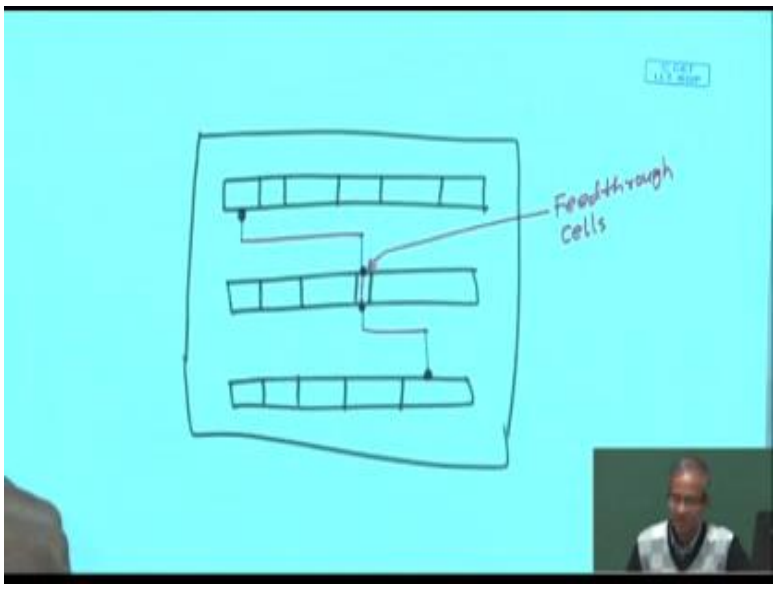


Let us see how I have a diagram here. This diagram shows one row of the standard cell cells. This is a second row; this is a third row. And, some interconnections are shown

like this. See when you do interconnection, as you can see these are shown as two different colors; which means the horizontal and vertical lines are on two different layers. So, when you want to connect them, you run a part on one layer and a part on other layer, so that this another connection that is going side by side does not intersect.

So, you see here you have one connection from here to here. And, there is another connection from here to here. But, here there is no interconnection, no cross connection because they are running on two different layers; blue and purple and, just another thing. You see you have this standard cells like this and this space in between which is there, you are leaving it for interconnections routing. So, here one thing is true that if you do it in this way, there may be situation where you need to connect a point from here to connect say some point here. Let us say you want to connect here to here. So, how do you connect because you see, let me just show it here.

(Refer Slide Time: 17:00)

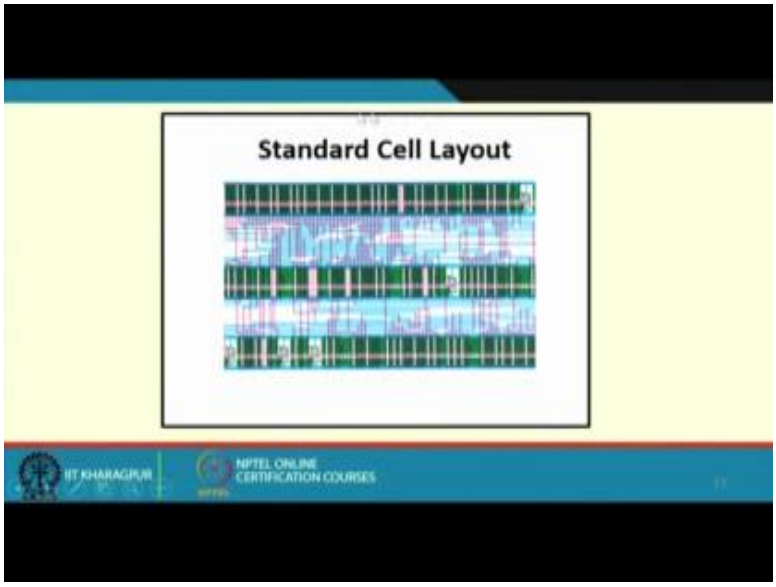


So, what I mean to say is that you have one row of cells here, you have one row of cells here, you have one row of cells here. So, this cells you are placing here one by one. Here also you are placing the cells, here also you are placing the cells. Now, suppose I need to connect a point here to a point here. So, how do I connect? I do not have any path to take from here to here. I have a path to take from here to here or here to here. So, what I do in this case is that I use some special standard cells, which are called feed though cells.

Feed through cells are nothing just they support a vertical connection nothing else, no logic.

So, if you want to make this connection what you do? Some, just showing using two different colors, you make a connection like this and from here you make another connection like this. These are called feed through cells. So, so here whenever required, you can include this feed through cells in your design, in your cells, so that you can take some wires from the top to bottom or bottom to top as required.

(Refer Slide Time: 18:51)



So, let us look at a more complex design; how a standard cell layout typically looks like. So, you can see that there are cells placed here, cells placed here, cells placed here and this space in between this is called a channel. This channel is used for interconnections. You see red and this pink wires are shown on two different layers. They are used for interconnection.

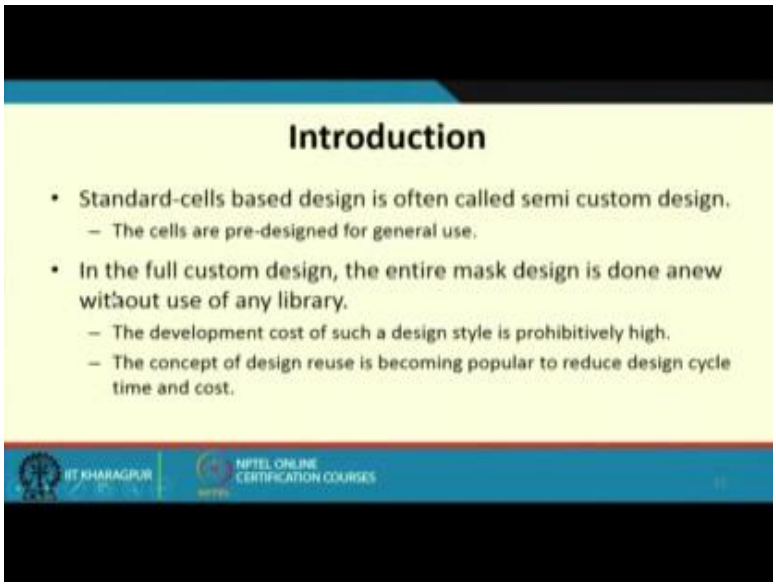
So, this is how a standard cell layout works. If you see the reason we are discussing the design styles, now will have a big impact on our subsequent discussion. Like, you see the first thing relating to physical design for VLSI circuits that we will see is a process called floor planning and placement. Placement means given a block, where in the chip I have to place the block. You see if you are following this standard cell design style, then your choice is very limited. You already have the rows defined. You can place a block in one of the rows. But, if you do not have any such restriction, then you can place the block

anywhere on the chip on any coordinate that is much more unrestricted. And, obviously it will be more complex.

So, it really depends on the design style whether you need to spend lot of time in certain steps or you do not need that at all. In standard cell, floor planning is not required, only placement, placing something. Floor planning means approximately where I will place it on the silicon floor. That is floor planning. So, this is how a standard cell layout looks like.

So, now let us come to the extreme full-custom design. So, here we are more saying that we want to design everything from scratch. But, as I said in practice today very really we do that because of the very high cost and time involved for that. But, anyway full-custom may mean means unrestricted. I can place anything anywhere on the chip. But, I am not saying that I am developing everything from scratch. May be for example, I need, let us say I need a multiplier as part of my design. But, already you have designed a multiplier. I can take it from you, take the design from you and I can put that same design in my design. I can reuse the designs in this way. So, people normally create their VLSI chips today in this fashion.

(Refer Slide Time: 22:00)

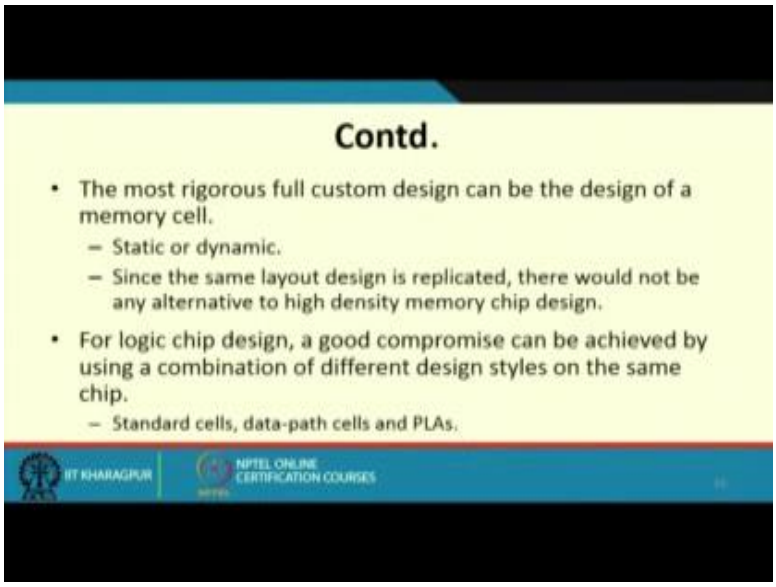


So, standard cell based design; as I said they are called semi-custom design because the cells are pre-designed. The gates, the multiplexers, the flip flops, they are of same heights and they are all pre-designed. But, in full-custom design we are saying that we

may not have any library at all. So, we may have to design all the blocks our self, we may have to place them in whatever location we want. We have entire flexibility. But, obviously for this entire flexibility to happen, the development cost can be prohibitively high. You think of the fact that modern day VLSI chips can contain more than a billion transistors. Doing it or handling it in a full-customer is almost next to impossible because it will take you years to create a design to complete your design, complete your layout. So, this will become too complicated for you. So, as I said I am emphasizing.

Design reuse is becoming a very important thing. So, whenever you have a design which was already done by some earlier designer, you will always try to reuse it in order to safe time, but there are something which was not there, you will have to design it from scratch anyway.

(Refer Slide Time: 23:37)

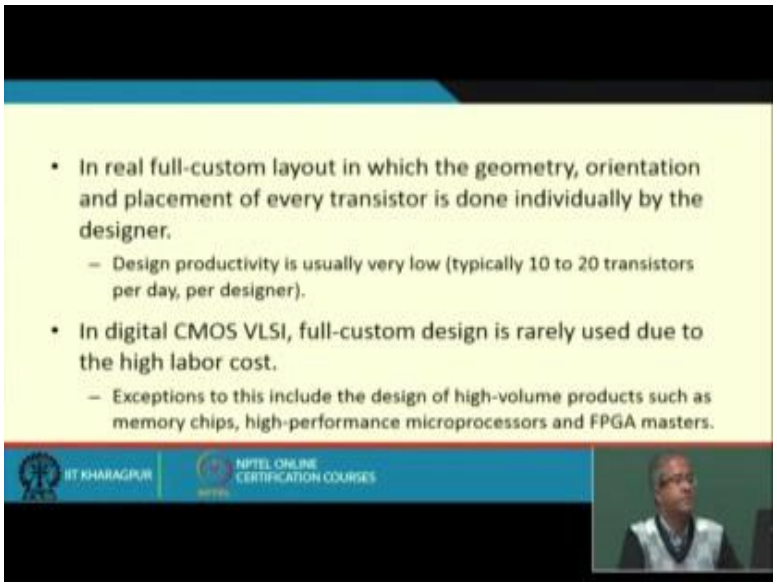


So, so one classical example where full-custom design is used very religiously and rigorously is in the design of a memory cell because memory is one kind of a chip where you try to pack maximum amount of cells or bits inside a chip.

So, very compact and optimums layout is of extreme importance and, one advantages of memory is that because it is created as an array, once you create the layout of a cell you can replicate it large number of times. So, it is not that you are creating the layout of the whole chip from scratch. You are designing the layout of a cell. You are replicating it many times, million number of times, billion number of times like that. So, for a memory

cell you can actually go for this. But, for logic chip design as I said, so you can use a combination of various designs styles and also you can use design reuse. Logic cell, you can use; some part you can use standard cell, some (Refer Time: 24:57) you can use; using design reuse you can include a block, like that.

(Refer Slide Time: 25:04)



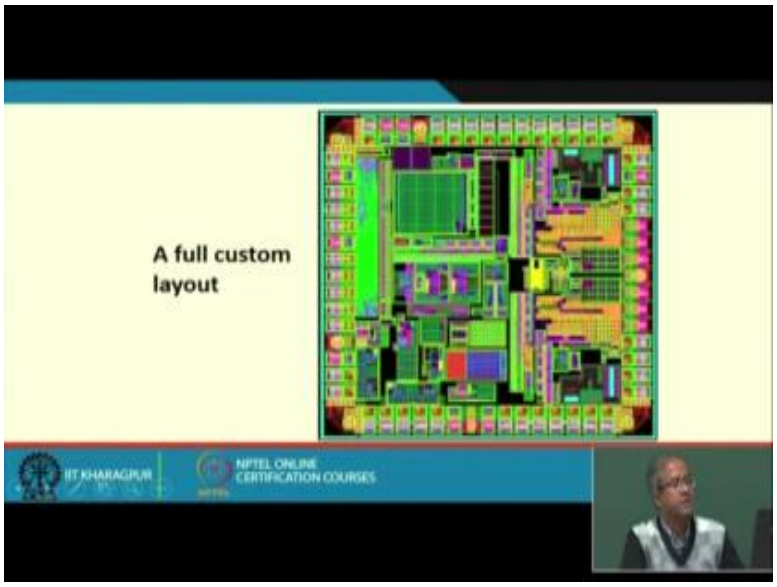
So, this is a typical figure, which will tell you why this concept of full-custom design will not work well; because in real full-custom design, so where, means everything is flexible; the geometry, the shapes, orientation, the angle of rotation, placement, whether place of every transistor of every circuit block can be done individually, the design productivity can be extremely low.

So, this was a average figure which is quoted 10 to 20 transistors per day per designer. So, for large designs it may not be practically visible to go for this kind of full-custom design. So, you see when you design some chips which we except to sell in millions; for example, the processor chips which Intel manufactures the Pentiums, the multicore chips, CPU chips, there they can afford to spend lot more time and effort in order to create a chip which is much better, much faster, much optimize. But, in other cases where we expected two sell 1000 or 10,000 of the chips in the market, doing or going through that exercise is not means worth the pain or the effort.

So, as I said in digital VLSI design, full-custom design is very rarely used. Exceptions as I mentioned are the design of high volume products such as high performance

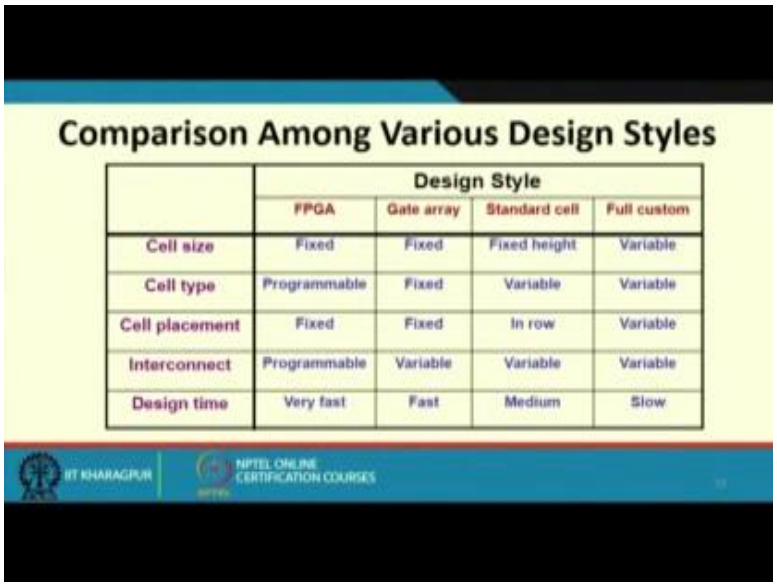
microprocessors which are expected to sold in plenty, memory chips and of course FPGA chips, which are also sold in very large numbers. It is here where some kind of full-custom design or some judicious combination of full-custom and semi-custom can be used.

(Refer Slide Time: 27:13)



So, a typical full-custom layout can look like this. So, you see you have different blocks of various shapes and sizes. Some are rectangular, some are long some are thin. So, here as you can see if here you cannot follow this standard cell kind of design principle, you will have to place this block individually on the silicon floor. So, after placing will have to interconnect them in a suitable way. So, the entire principle, the entire philosophy of design here is different. We have to do it in a different way.

(Refer Slide Time: 27:56)



So, in order to make a quick comparison among the design styles, we are comparing FPGA gate array, standard cell and full-custom against these parameters. Cell size: for an FPGA we have the CLBs, they are fixed; for a gate array they are the transistors or the gates which are fixed; standard cell, cell size can vary, but their heights are fixed. But, for full-custom it is entirely variable, entirely flexible.

Cell type: for FPGA you have this CLBs which can be programmed; for gate array, it is transistor or gate which is fixed; for standard cell, you can put in any cell from the library; full-custom also you can put anything variable.

Cell placement: FPGA gate array is you do not have any choice. They are already pre-fabricated. They are fixed. In standard cell, you can place the cells only in rows, but in full-custom you can place them anywhere.

For interconnects: FPGAs, they are programmable switch matrix which you can program; gate array, you have to complete the interconnection. This is variable. Standard cell, full-custom, they are also variable.

And, obviously design time wise FPGA is the fastest. It can take minutes to an hour, may be in the lab. Gate array is relatively fast, standard cell is medium, full-custom is slow. See, medium means you need turnaround time of 6 months to 12 months. That is what we call as medium. But, full-custom complete design will require much more time.

So, with this we come to the end of the fourth lecture. Now, in the next two lectures we shall be looking at some brief overview; means on about the different steps in the VLSI physical design cycle, which will be covering in the next modules in this course.

Thank you.