VLSI Design Flow: RTL to GDS

Dr. Sneh Saurabh

Department of Electronics and Communication Engineering
IIIT-Delhi

Lecture 53
Post-layout Verification and Signoff

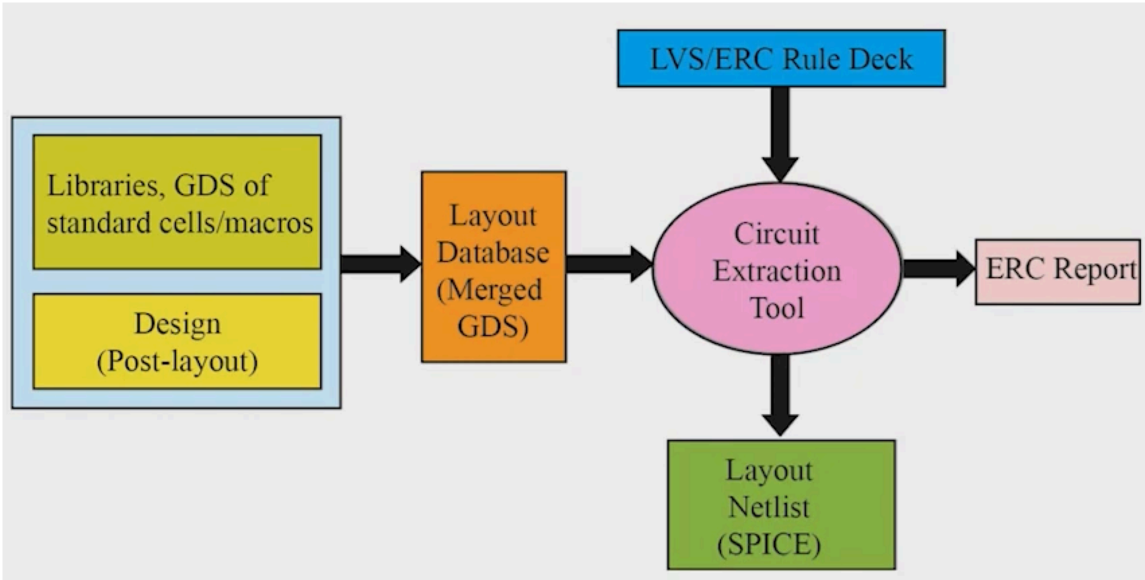
Hello everybody. Welcome to the course VLSI design flow RTL to GDS. This is the 42nd lecture. In this lecture we will be discussing post layout verification and sign off. In earlier lectures we had looked into various physical design tasks such as chip planning, placement, clock tree synthesis and routing. So, at the end of detailed routing we have basically created the layout which is complete and this layout can be used by the foundry for creating the chips or manufacturing the chips.

However, before sending the final layout or this layout that was created after detailed routing to the foundry we do some post layout verification to ensure that our layout is indeed carrying out or delivering the functionality that we wanted and also the figures of merit are acceptable. So, in this lecture we will be looking into what are these post layout verification tasks that we will need to carry out after detailed routing and before sending the final layout to the foundry for fabrication. Specifically in this lecture we will be looking into layout extraction, physical verification, eco or engineering change order and sign off. So, first let us look into layout extraction.

Now what is the motivation of layout extraction? So, once when we have created a layout, basically the layout is nothing but various shapes or polygons that we need to or the shapes or polygons that need to be fabricated on various layers during fabrication or during photolithography. So, the layout is composed of various shapes or polygons. Now the various verification tools such as static timing analysis tool, signal integrity tool and other physical verification tools if they work directly on the layout or directly on the polygons that are on the layout it will be a very difficult or challenging task for them to do the computation and carry out the verification. Therefore what we do is that once our layout has been created we have the final layout has been created after detailed routing then we extract various information from the layout using the layout that was created we extract out some information the useful information that are relevant for post layout verification and then give it to the tools various tools for example, tools that do the timing analysis, power analysis and other types of analysis that are needed in the post layout verification task. Those take the information or they take the extracted information and do the analysis and with the help of the when we get the extracted information rather than the complete layout then the computation becomes easy for these post layout verification tools and this is the primary motivation of carrying out layout extraction.

Now, layout extraction consists of two major tasks and what are these tasks? The first is circuit extraction or devices and interconnection interconnections that are extracted from the layout right now. We have created the layout and layout consists of polygons. Now the tool the layout extraction tool will first extract out various kinds of devices devices meaning transistors and other things that have we have actually created on our layout or we intended to create on the layout and the interconnections meaning the nets that we created during detail routing or say during power the powers grid that we created or the clock mesh that we created and so on. So those interconnections and the devices are first extracted during the circuit extraction step. Then in the next step what is done is that if the parasitics are extracted right. So the parasitics are various resistances capacitances and inductances that are associated with the layout, but which we did not intend to create that they get created on its own right for example, coupling capacitance between two wires right we we do not intend to create those capacitors, but those capacitors are created automatically or those parasitic capacitors exist in the layout and that information also needs to be extracted by the tool.

So that say single integrity tool can understand can comprehend that how much the capacitance is between two lines and based on that it can check whether signal signal integrity issues is there in our layout and do we need to fix them and so on right. So now let us look into these two tasks that are circuit extraction and parasitic extraction in some more detail. Now in this figure we are showing the framework on which the circuit extraction tool works right. So this is that circuit extraction tool right the circuit extraction tool and it takes these two inputs right the inputs are coming here right. So what are the inputs? So the first input is the merge GDS right.



Now what do we mean by merge GDS? So once we created our layout right when we did the when we created our design and and carried out say the routing the the detailed routing also then our then our layout was created in the layout at this stage what happens is that we have an abstract abstract view of the cells right. For example, the cells for example, an inverter or say AND gate these are the standard cells. Now the view of these standard cells is an abstract view and these views are based on say what is the view on the left side right. So remember that when we carry out physical design we give a library which is one of the formats in the left format and the left format consists of a star extract view of the cells. So it knows where the prints are, what the bounding box is and so on.

But it does not know the exact layout of the inverter for example, if this is an inverter then the exact layout of how many a pmos are there how the n mos are there and how the connections are made on the layout internally in this inverter that is not there. That information is actually present in the gds file or the layout file of the standard cells and macros and those are those are also those will these files will also come from the pdk, but then we have to merge these two information right. So, the design or the post layout design contains an abstract view of the inverters and how the various kinds of standard cells are connected and so on. But the internal detail of the inverter layout that is there in the in the library or in the in the gds of the standard cells cells which is contained in the pdk we need to merge these two information and then the merge g d s is basically the the final layout that can be used for for for fabrication right. Now physical design tools have capabilities to simply merge these two information and they will create a that they can give us a merge g d s which we can give it to the circuit extraction tool right.

Now circuit extraction tool what it will do is that it will take the information of the extraction rules using say lvs or erc rule day right. So, then what is a rule day? So, rule day basically is a set of instructions that are coded in some format in a format. So, this circuit extraction tool can comprehend how to extract various kinds of devices from the layout. For example, the rule can be that if there is a poly and it is intersecting with a diffusion layer then that is an indication that there is a transistor at the intersection right. So, similar kinds of rules will be coded or instructions will be coded in in the rule deck and using those instructions or the extraction rules that tool will be able to extract various kinds of devices from the layout right.

Now these extraction rules will come from the foundry and it will differ for different technologies. For example, say for 22 nanometer it will be a different set of lvs or erc rule deck or for say 7 nanometer it will be quite different right. So, these the information that how to extract the devices from the layout that comes from the from the ruled from the from the foundry and it is encoded in the in a in some format that the tool can understand and that and the file that we use in our design flow those are known as rule deck. So, in this case the lvs rule deck or erc rule decks are typically used for circuit extraction right. Now what will be the output of the circuit extraction? So, the first output will be the layout layout netlist right.

So, what layout netlist means is that from the layout the extraction tool will create a net list. For example, if this is say an inverter and this inverter is connected to another inverter like this right on the layout then then the tool will extract that information and and and and and generate a net list which is typically in the spice form right. So, in the spice form it can describe how many transistors are there, how they are connected and so on right. For example, if this is an inverter right this is one inverter I1 this is another inverter I2 right. Now we know that an inverter pCMOS inverter consists of a pMOS and an nMOS right.

So, we will have something like this some structure like this we will have nMOS pMOS nMOS right and this will be the crown line and then this is the power line and this one the this is the output of this inverter this inverter right this is the output being fed to the next inverter right. So, this is the next inverter it will be fed to this right. So, the circuit extraction tool will basically generate a net list of this form and will be typically generated in a format of spice right. So, this is what the layout netlist is, meaning that this net list has been created or extracted out of the layout that was given to the tool. Additionally the circuit extraction tool can also give ERC report right because during such circuit extraction the rule deck contains the ERC rules that are the connectivity rules or or or the rules for having valid connectivity.

Then the tool can also find out what are invalid connection rights and those can be reported during the circuit extraction. Now, what is parasitic extraction? Now, parasitic extractions are basically the unwanted resistance capacitance and inductance which are created on our layout right. We did not want intentionally to we did not create these resistors, capacitors and and inductors in our layout, but they are automatically created because of the way we make the interconnections and so on right. So, in the parasitic extraction the what the parasitic extraction tool will do is extract all these resistance capacitance and inductance values from the layout and report it in some form right. So, the resistance is extracted for each net in the on the layout and while extracting it what the tool can do is that probably take a consider net and then segment it into various sections right.

And for each segment the tool can estimate or compute the resistance based on its sheet resistance length, weight and so on. And then report the and combine the result of various segments to get the resistance of the complete net and then this process will be repeated for all the nets in our design. Now, capacitance extraction needs to be done such that all the various components are the components that we discussed in our earlier lectures. So, if there are wires right and there are other wires in its vicinity then there are various kinds of coupling capacitances and there are and and these and the coupling capacitance can also be of various types right. We have seen that the coupling capacitance can be because of fringe because of lateral lateral proximity between two wires or because of the overlap between two wires and so on.

So, all these components need to be considered while capacitance extraction. So, in general capacitance extraction is a more difficult problem than resistance extraction at what what tools typically do that at the chip level when we have lots of nets in our design or in the layout and we need to compute the parasitic capacitance for all of them the chip level parasitic extraction is decomposed into two tasks right. So, what are these two tasks? The first is technology pre characterization and the second task is pattern matching right. Now, let us look into these two tasks more carefully. So, what is technology pre characterization? So, technology pre characterization is performed only once for a given technology maybe one when say seven nanometer technology is being developed then we have to do pre characterization of technology pre characterization for that given technology and and subsequently the information that comes out from pre character pre characterization for a given technology it will be used for many designs right.

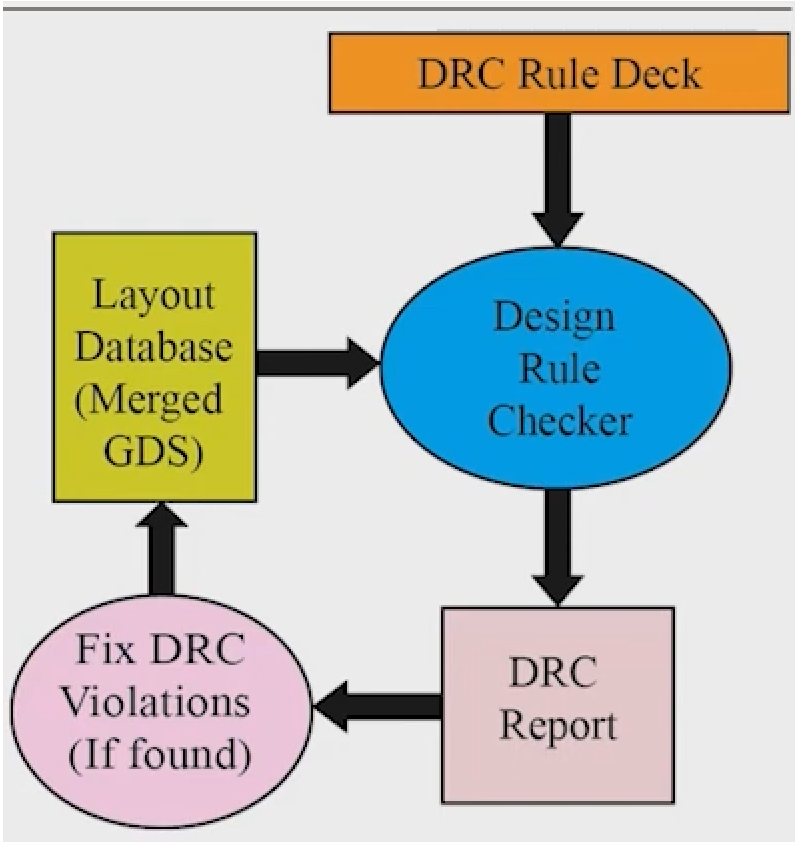
Now, what is done in the pre characterization step? So, in pre characterization what the tool will do is that it will enumerate millions of sample geometries and structures right. Now, how are these millions of sample geometries and structures created or for a given technology? So, it comes from the stack of the interconnects and the devices that can be there in for in a for a given technology right. So, based on which the different stack of interconnects layers and the and the dielectric the tool will come up with various combinations that can be there between different metal layers right and also various kind of in addition to various combination the tool will try try try out various various geometries right or different different kind of spacing and which which in which which conductor or which kind which metal layer is in proximity with other metal layers and so on right. So, based on the gate stack or sorry the technology stack and the interconnect stack the tool may come up with a one one one one combination like the metal one is there m one and say m two is just above it m two is just above it and say another m two layer is somewhere here right. So, this is one configuration that the tool may come up with right.

Now, similarly there can be many such combinations and different kinds of geometries and configurations of the metal layers and and the tool will enumerate them and can be millions of such such combinations can be there and for each of them the tool will what the tool will compute the capacitance using more accurate field solvers right. Now, what are field solvers? So, field solvers do numerical computation right. So, it divides the structure into measures and does a numerical computation and computes the capacitance very accurately using sophisticated techniques. So, using this field solvers what the the tool will compute the capacitance for various configurations or different geometries and structures and then it will put the capacitance value either in a look up table or using the capacitance value computed for various structures it can come up with some empirical formula created using curve fitting right and that empirical formula will be stored during the pre characterization step. Now, thispre characterization characterization task is very time consumed, but fortunately we need to do it only for once for a given technology and this information will be utilized by all the designs that will be fabricated using this technology and therefore, the cost in with or effort gone into technology pre characterization that gets distributed or get amortized over multiple designs and the effective cost or cost of of of technology pre characterization comes comes down right.

Now, the second task in parasitic extraction is pattern match right. Now, this pattern matching is layout specific right. Now, what is done in pattern matching is that we partition a layout into smaller windows right. So, if there is a big layout it will be partitioned into small small windows right and then match windows with pre characterized part partitions right. So, suppose this is the window in our layout this is our layout then this this this window will be matched with with some other window some pre characterized configuration and if it matches right then the it will compute the capacitance with the help of look up table or empirical formula that was stored for that configuration during the pre characterization step right.

Now, during this computation the tool will use the actual geometries of the layout then let us now look into what physical verification task we need to carry out after creating the layout. So, there are three major physical verification tasks that we need to carry out. The first is design rule check or DRC the second one is electrical rule check or ERC and the third is layout versus schematic check or LVS. Now, we will look into all these three tasks in some detail. Now, what is the design rule check? So, design rule check basically ensures that the layout meets the constraint required for manufacturing and these rules are basically defined by the respective foundry. So, we have for example, created a layout for say 14 nanometers then we will use the design rules that will be coming from the foundry where we want our circuit to be fabricated.

So, it will be coming from the foundry that has actually given us the lab libraries PDKs for 14 nanometer based on which we have designed right. So, the PDK will also contain the design rules that need to be checked right. So, why do we need to or why do foundries try to enforce these design rules? So, the basic motivation is that if these design rules are followed or those are those of this design or we ensure that these design rules are honored in our layout then we will be able to achieve good yield and also improve the reliability of our circuit right. So, these design rules can vary with the technology and typically what happens is that as the technology progresses the feature size decreases and at smaller feature size these rules become more and more complicated. So, this figure basically shows the framework of design rule check.

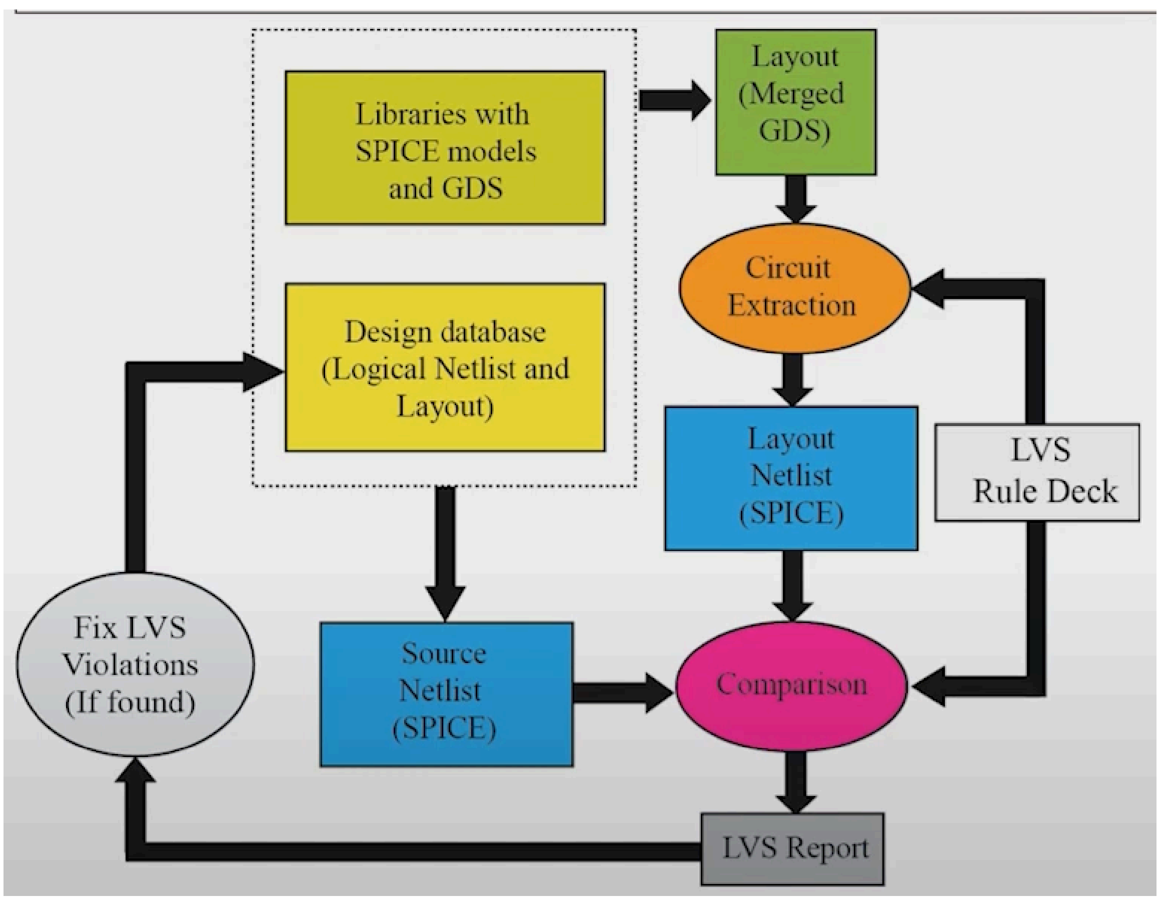


So, for design rule check. So, this is the basic tool that carries that that performs design rule checking right. Now, what are the inputs? The first input is the layout database right, that is the merge GDS that we discussed in previously and then these are DRC rules. So, this will contain basically the the the rules that needs to be followed in the layout right and using those in these two information the design rule check will design design rule checker will generate a report or DRC report and if we find violation then we need to fix those violation right and this has to be done iteratively until we have fixed all the violations design rule violations. Let us look into a few examples of design rules for example, there can be a design rule which says that the minimum width of poly should be a given value say w right. Now, if in the layout the width of this poly is smaller than this w right then it will get violated, it will be flagged and it will come in the DRC report right.

If and now to fix this what we have to do we have to increase the width of this poly right. So, that it becomes more than w right. Similarly, there can be a rule that there the spacing between two poly should be at least S capital S. Now, in our layout if this spacing is less right then the tool will flag that there is a violation of minimum spacing right and this will be there in the DRC report and then looking at that we will need to fix it for example, in this case we have to make the spacing larger more than the defined capital S right. Then once we have created the layout we also need to check for ERC or electrical rule check.

Now, what electrical rule checks do is basically check design for electrical connections that can be problematic right. So, the connections which can be problematic are flagged and then we have to fix those things right. For example, the problems can be the shorts, open, floating gate, floating nets etcetera right. So, when we do say circuit extraction probably at that time if the rule deck contains ERC rules also then ERC report will be created at that time and then we can fix without going into further down the flow we can we can just look into the where this the rule violation was there and we can fix it. Now, what is LVS? LVS is basically layout versus schematic check right.

It verifies whether the layout corresponds to the original schematic net list of the design right. So, we started physical design with the net list right. So, after logic synthesis we got a net list and then from using that net list we went through a physical design task and at the end of that we have our our layout right. Now, we want that the layout basically delivers the same functionality that our original net list was delivered right and that is what the basic purpose of layout versus schematic check is right. Now, let us look into the framework of layout versus schematic check.



So, this framework looks very complicated right, but it is not that complicated. What layout versus schematic check is doing is that basically it is doing the comparison of two entities right. So, this is the basic task that is being performed by LVS right LVS. So, it is doing a comparison of two net lists and what are these two net lists? The first netlist is the extracted netlist right. So, the layout net list that we extracted remember that when we discussed circuit extraction we said that given our design and the and our and the and the libraries and the pdk which contains the gds of each individual standard shells and macros we created a merge gds and using merge gds we did a circuit extraction and did a circuit extraction right and the circuit extraction when we did we get the layout netlist right. So, this portion is the same that we discussed in the layout extraction right.

So, the layout net list that we got from layout extraction is compared with the source net list. Now, what is the source net list? So, source net list is the logical net list combined with the device information right. The source net list contains the information of the net or it is it is it is the same net list that we started the physical design with right. So, after logic synthesis we got a net list right. Now, that net list was consisting of that say how the net and gate sorry and gate was there and and gate was there and inverter was there how they were connected and so on.

So, that information is in the logical net list right. Now, with the logical net list we combine the device information right. For example, if we have say an inverter right this inverter internally contains a n p mos and an n mos right. So, this information is also contained in the pdk that each standard shell consists of which all transistors and devices and so on. So, that information is again taken and merged with this logical net list to get the source net list.

And these two source net list and the layout net list these two are compared if the comparison is successful right. Then we are we are done with that if the LVS report is there and it shows some violation right that if there are some discrepancies then we need to go back to our to our layout or our design and see that whether what is the problem and then probably fix it until the layout versus schematic check passes right. Then now let us go into the sign on right now we know after we have created the layout right. Now, we want to basically do post layout verification in a thorough manner and sign off only after all the checks are passing right. Now, what is sign off? Sign off is a series of verification steps that must be carried out before sending the layout or GDS to the foundry right.

Now, why do we sign off checks? We do the sign off checks because these checks ensure that the layout delivers the intended functionality and also meets the various figures of merit that we intend right. So, let us look into the framework of sign off check right. So, in the sign off check what will be done is that we have the design layout this layout was created say after after after a detailed routing and then from this layout we extracted design design information we saw what the extraction layout extraction is and using layout extraction we created we got various information like what are the parasitic, what is the net list and so on right. Now, using both that information a series of checks will be done right. So, here I am showing some of the checks like static timing analysis will be done then physical verification task will be done LVS ERC DRC that we just discussed right and then there can be a signal integrity check right.

Now, note that once we have created the layout of the interconnects then only we can extract the coupling capacitance and then only we can do the signal integrity checks right more precisely or accurately. So, after we have created the layout we do the signal integrity checks we also do power integrity checks for example, say it can be related to IR drop in the power lines or it can be related to electromigration checks and so on. So, after creating the the layout of the interconnects carrying out these checks becomes a more accurate or more reliable and therefore, after we have created the created the layout we need to carry out these checks again even though we have might have carried out these checks earlier in our design flow right because the accuracy at the this stage is much higher than the accuracy we had earlier. Right and there can be other checks which are which can be defined by a design house or the company which is designing or a designer can also say that these are some of the more extra checks that we need to do before sending it to the for the tape out or for the for for the for before sending it out to foundry for fabrication. Now, typically what is done is that these sign off checks are done using a separate set of tools that we use during the implementation right.

So, in the implementation phase we use some tools whose major purpose is to implement our design right. They also do some checking for example, related to timing with what is related to say DRC and other things. But the accuracy of those checks are not not or the or the they do these checks at a higher level of abstraction and therefore, the accuracy of those checks can be lower right therefore, once we are going to sign off we take another set of tools which are separate from the implementation tools. So, though we take help of sign off check sign off checking tools which have got much higher accuracy for carrying out these kinds of verification right. Now, if we find violations at this stage right now we have done everything, but we are getting a few violations right. Now, how to fix those violations right now if we want to fix those violations we might need to change our design right.

So, therefore, we again go into the into the in into the implementation phase right. So, if the if we are lucky then we can basically make changes in using ECO what is ECO we'll just see or if we are not that much lucky then we might need to go back and fix issues say in chip planning if we are highly unlucky or in placement or in clock trace interties or in global routing or in detail routing right. So, if the fixes are localized or as close to the end or end of the design flow then the cost and the effort for fixing will be less right. If we find a problem which is deep rooted at say very early in the design flow then the cost of fixing it is much higher right. And therefore, at the sign off stage at the sign off stage will create loops in our design flow right. So, this can create loops in our design flow and we might have to carry out these design tasks that I have listed in this figure multiple times before the design closure is achieved right.

Now, if there are localized changes and liberalized changes then we can take help from ECO and fix those changes or do those changes to our design. Now, what is ECO? ECO stands for engineering change order right. So, this is a mechanism which is through which we introduce controlled changes in our design right. So, why do we want to or when do we need to carry out these controlled changes or ECO changes. So, sometimes at the last moment incremental changes are needed in our design right.

And these incremental changes can be because of a bug that was discovered very late in the design flow or a some functionality change request came much later and when we are in the design flow and we are forced to make those changes we cannot avoid those changes right. So, if such kind of changes are there in our design which are coming at the last moment of the design flow then we have to be very careful right why because these changes can be very risky it can introduce new errors and therefore, we incorporate these changes using engineering change order right. So, we do it very carefully through the engineering change order mechanism and it is at the end of this design flow right we do these changes and we take help of what are known as ECO tools. Now, what do these tools do? So, these tools enable making targeted incremental changes rather than re implementing the design right. So, if that if we want localized changes then these ECO tools help us to make localized changes in our design in a more efficient way and and and in a better way why does it allow it as a better way do allow us to do these changes in a more efficient way.

These tools not only make changes in our design, but they also help us verify the correctness of the ECO changes. For example, internally it can run a formal tool or timing analysis tool or those kinds of tools which will ensure that the behavior of the functionality is not disturbed for our design while making the changes. And therefore, it saves these ECOs tools save design designer time, effort, cost and also it minimizes the risk. Now, there are two types of ECO changes, primarily two types of ECO changes, the first is functional ECO. In functional ECO we are changing the logic right, we are changing the functionality of our circuit. Maybe we want to replace an AND gate with a NAND gate or so on right.

So, though that means that there is a functionality change right. So, if it can require logic re-synthesis or we can use the spare cells right, remember that in the placement stage we put some spare cells and also in our design right. So, if we do not want to go all the way up to the logic synthesis we want to make changes only in our physical design or in the interconnections we can take help of spare cells to carry out these functional ECOs. And there can be some direct changes in the layout to fix setup hold violations, SI related violations and design rule violations. So, these violations will be discovered at the sign off stage right.

So, we saw that in the sign off stage we carried out multiple verification tasks. Now, those tasks can expose those bugs. Now, to for fixing those bugs if those bugs are localized in a small area we can take help of this ECO tools and that will help us ah ah ah reduce our designers effort and also we can also ensure the correctness of the of the of the of the changes with the help of the ECO tools right. So, we will carry out this sign off stage sign off verification task. So, here ah ah here a list of tasks is shown right and, but do not take it that these tasks need to be done in a sequential manner.

Typically when we go to the sign off stage we are very very close to the deadline of or of ah completing our project or doing a taper right. So, typically during that time it is a kind of ah ah fire fighting scenario in the design house right and in that case typically these various sign off checks can be paralyzed given to various groups which do carry them out in parallel. So, that the design can be closed in a shorter time right. So, once we are free with all these all the all all these verification task shows that there are there is there are no issues in our design or the issues if they exist those are tolerable then we go to the next step and next step is the design tape out right. So, what is design tape out? Design tape out means that we send the GDS for or GDS or the layout to the foundry for fabrication right.

So, now, this design tape out is a kind of a time for celebration for the designers right because the arduous design activity that started with say system level specifications and then finally, culminated in the GDS being sent to the foundry that concludes the project right. And therefore, this is an occasion to celebrate for the designers and it is also an occasion to celebrate for us because we have come to the end of this course and we have completed our journey from RTL to GDS. Finally, since this is the last lecture let me summarize what we have done in this course. So, in this course we started with taking an overview of VLSI design flow and then we looked into logic design both implementation and verification. Then we moved to design for a test or DFT and finally, we moved to a physical design task and then physical verification and signing off right.

So, in this course we have covered the entire design flow from say from the RTL to GDS right. So, the basic objective of this course was to cover the breadth of the VLSI design flow which we have done fairly well. Now, since this course was only 12 weeks right. So, at some points we have sacrificed the depth, but for the foundational course this is acceptable because we have laid a very strong foundation for this course during these 12 weeks and over this you can build your expertise on your own. Now, one more thing I would like to highlight is that the VLSI design flow is a course where practical skills are very important right.

So, to gain those practical skills you have to run the CAD tools, analyze those results and then understand what goes inside what is the correct input, what is the wrong input and so on right. So, in this course we have given a few tutorials on these open source tools, but I think that you will be able to learn better if you run these tools on your own analyze on your own and then that will give you a lot of confidence and then you can become an excellent VLSI designer and you will become an asset to the semiconductor industry. Now, if you if there are any feedback any questions I am always reachable I will be happy to get your response you can tell me what went good in this course what can be improved and if you have any questions you can free to reach out to me right. So, I think various kinds of students and participants are there in this course. I wish all of them very good luck and in their future endeavors and a successful career ahead. Thank you very much.