VLSI Physical Design with Timing Analysis

Dr. Bishnu Prasad Das

Department of Electronics and Communication Engineering

Indian Institute of Technology, Roorkee

Week 05

Lecture 24

Introduction to Floorplanning

Welcome to the course on VLSI Physical Design with Timing Analysis. In this lecture, we will discuss about the floor planning. So, the content of this lecture includes introduction to the floor planning, and we will discuss about what is the difference between floor planning and placement. Then we will also discuss about the different objectives of the floor planning, then optimization goals. There are several terminologies are involved in floor planning that will also be explained in detail. So, in case of a VLSI design flow, we have the first step is your partitioning what we have already discussed. In the next step, we are discussing about the chip planning or floor planning. Chip planning and floor planning is the same thing. This is the second step of your VLSI physical design flow. So, basically if you can see the partitioning divides your design into individual circuit modules, but we need to determine many more things.

What are the things? Basically, the shape of each block and location of each block and the location of the pin. So, we need shape of the blocks, location of each block and location of pin on the boundary of the block. So, all these things are also essential and need to be determined. So, the floor planning phase of your VLSI physical design basically performs all these jobs. So, you have different types of blocks are there.

One is called hard blocks or the hard IP and some there are another category of the blocks is called the soft IP. So, hard block is one category, and soft blocks are the another category. What is the difference between hard blocks or hard IP versus soft blocks or the soft IP? In case of hard blocks, this area is fixed, and dimensions are also fixed. For example, a memory generated by the memory compiler. Example of hard IPs are memory generated by memory compiler.

So, that is determined by the memory compiler. We have some options might be, but it is not flexible, that much flexible. So, in case of hard IP, let us say if I want to use a PLL inside my block, PLL or ADC inside my block, I want to use that ADC provided by third

party in my top-level chip level integration. In that case my PLL or ADC that the area and dimension of that block is fixed. So, those are called hard blocks. We cannot have any flexibility to play with the hard blocks. But in case of a soft blocks, area is fixed, but we have a flexibility of adjusting the height and width. Area will be same, but we can change the aspect ratio. So, those blocks are the blocks we can change during the floor planning stage. So, there are two phases are there, one is called floor planning and second one is the placement.

In case of floor planning, it consists of planning and sizing of the block and interconnects. But in case of a placement, we have specific location is assigned to the block, but in case of floor planning, it is a relative placement. The placement of the block is not fixed, but after the in case of floor planning, the placement of the block is not fixed. The coordinates are not exactly assigned, it is relative. But in case of a placement, the coordinates of the blocks are assigned. The interconnection of the blocks is completed in the routing phase. The interconnection of the blocks is completed in the routing phase. So, in case of placement problem, we are assigning the location of the hard blocks on the surface, layout surface is called your placement problem. If some of the blocks are flexible, then the problem is called the floor planning problem. So, we can change the shape of the block and we can put it. And your placement is basically restricted version of the floor planning problem. Placement is a restricted version of the floor planning problem. So, we will see what is the input to your floor planning problem, what is the input to your placement problem. So, first we will discuss about the floor planning problem. So, in case of floor planning, what are the inputs we give to that algorithm.

So, we have set of blocks, area of the each block, we have set of blocks, we also know the area of each block and also all possible combinations of the block means let us say I have block A, it can have different orientation of the block. So, all possible sets basically the orientation of the blocks are given. So, then the fourth one is basically number of terminals of each block, number of pins of each block that is also given. And the netlist, how the blocks are connected with each other, the interconnection between them that is also provided as input to your floor planning phase. In case of placement, we have a set of blocks with well-defined shapes.

Its shapes are already determined, we cannot have any flexibility to play with the shape of the blocks because it is already determined in case of a floor planning stage. Then we know the pin location, so we also provide the pin locations as input to the algorithm, placement algorithm and of course, the connection between the blocks need to be provided through the netlist, how the blocks are connected with each other that is also provided in case of a placement phase. So, we have less number of input in case of placement, we have more number of inputs in case of a floor planning. And one more difference is that output of floor planning is going as input to the placement stage, output

of your floor planning stage which defines the shape of your block is going as input to your placement problem. So, let us look into this diagram.

Here we have several modules are there, module A, module B, module C, module D, module E are there. We have a design, so this is the design we want to implement, and we have some power supply IO, we have some other IO pins are there. Now we have to determine how do we place this block in this chip area, we have to determine how we place these blocks in this chip area such that I can do the routing all these things in a proper manner. So how I can determine that one we will discuss in this lecture actually, we will discuss that in case of floor planning. So, what are the objective of your floor planning? There are several objectives are there, so it basically determines the shape and location of each of the blocks which will useful for gate placement and also it also used for pin assignment.

What is this pin assignment is actually? Every pin that has a external connection, so the pins let us say I have a block here, I have another block here. So, I have pins here, I have pins here. So, the locations of the pins are determined such a way that routing distance between the blocks that the interconnect length should be minimized. So, the pin location so is determined such that the input connection is connected with the output world, output blocks such that your interconnection distance will be minimized. So here we are giving an example here, we have block A which has different orientation, area of the block is basically is 4. So, area is preserved, we do not change the area but we can change the shape of the blocks. In this case your width is 1, the width of the block is 1 and height is 4. Now in this case my width is 4, height is 1. In this case my width is 2 and height is 2. In all cases my area is 4. So now I have another block, so here the width is 3 and height is 2. And the area is 6, here the width is 2 and height is 3. So how many blocks I have? A block has 3 different orientations, B block have 2 different orientations. So, I have another block is there, the C which is basically has width 1 and height is 2, here the width is 2 and height is 1. So now we need to determine how I can place all these blocks such A, B, C block in one of the chip layout such that my area is minimum.

This is the your floor planning problem. So, we have 3 different combinations are there, we need to check that what is the area of each of the combinations. So, if you can see here this is basically your 3 and this is 1 and this one is 2 and this is 1. So overall bounding box area is this is 4 and this area this is basically 3. So, your width of the bounding box is 4 and the height of the bounding box is 3.

So, then we have to look into other combinations. So here if you can see the width is basically 3, height is 2 and this is 2. So, your width is 3, height is 4. Now if you can look into here this is 1 and this is basically 2, now this one is 4. So now this one overall bounding box width is 3, so we have width of 3 and height of 4.

So, what it tells that even if the area of all the floor plan is same sometimes some of the orientation is suitable for some design. So, if you have 3 by 4 or 4 by 3 we need to check either of them or basically this is one category, these two are in one category because 3 cross 4 this is one type of floor plan and this is another type of floor plan. So, depending upon the requirement of your chip area we can place these blocks in the chip. Now there are several types of floor planning algorithms are there we are just putting all these together in one of the slides. So here if you can see your constraint based methods are there, we have integer programming based method, we have rectangular dualization based method, we have hierarchical tree based methods is also there, we have simulated evolution algorithm and the last one is basically timing driven floor planning algorithm.

So, there are 6 different algorithms are there, so this is a floor planning is quite useful while designing a chip because it will optimize your area of your overall chip design. So, what are the optimization goal actually? So, optimization of the goal of your floor plan is to optimize your block location aspect ratio using some simple objective functions. So, the main key objectives are how to minimize the area of the global bounding box. The global bounding box if you can go here the bounding box which will cover this one that is your global bounding box. So, this the boundary of the floor plan will determine the boundary of your boundary of this floor plan will determine the boundary of your design which will be placed in the top level. So, we need to minimize the area of the global bounding box which will optimize your aspect ratio. So, then another objective is basically total wirelength because your speed is critical, your speed of your design is critical for the operation of your design. So, we need to optimize the total wirelength, optimize means minimize. So, it is a minimization problem basically it is a minimize the total wirelength to reduce the propagation delay and power consumption. So, if you have less interconnect your signal propagation delay will also be less.

Then the area then first goal is basically area and shape of your global bounding box. What it says that the global bounding box of a floor plan how it is defined is the minimum axis aligned rectangle. Axis aligned means it should be aligned with the x and y axis. It is the x here this is the x axis, and this is the y axis. So, it is the axis aligned rectangle enclosing all the floor plan blocks.

So, the bounding box which is aligned with your axis the overall area comes under the category of global bounding box. Area of the bounding box is directly impact your circuit performance, yield or the area of your bounding box it will directly impact the overall speed of your design, manufacturing cost and finally the yield of your product also. So, the prime objective is to minimize the bounding box by finding the x y locations and aspect ratio of the individual blocks. So, another optimization objective is to keep the bounding box and aspect ratio close to given target value. So, you have a objective function we need to objective whatever is given target value whatever is given by the designer we need to achieve that global bounding box within that bound.

Then we can achieve the goal of your overall design. So, then we have total wire length so if long interconnection between the blocks will increase your propagation delay. So, your wire length needs to be reduced and which will also reduce your power consumption and manufacturing cost. So, the overly dense connections can lead to routing issues. So now we have two objective functions, one is area, one is wire length.

So here what is we are, finding a combined objective function where we have given priority to each one of them. So, the priority is given by this alpha which lies between 0 to 1. So here my area is important for me, or length of the wires are important for me based on that I will assign the value of alpha. If I have equal priority to both of them the alpha should be alpha should be 0.5. If I have more priority to the area less priority to the length of the wire length so then alpha will be actually close to 1. So, in that case alpha will be close to 1. So based on your requirement you can create a combined objective function taking area and wire length into account. Then the signal delay basically the interconnect delay has it is also very crucial which is also determine the critical path and critical path of your design. So, the static timing analysis is exploited to identify the critical nets critical interconnect paths which lies and we need to optimize those paths.

So, every stage of your physical design we need to run the STA to check that which are the paths are getting critical and we need to focus more on those paths to make them less critical in the future phase of your physical design. So, we have some terminologies which is used in flow planning we will discuss those terminologies. One of them is rectangular dissection. So, we have the rectangular dissection basically divides the total chip area into set of blocks or non-overlapping rectangles. So, you have a overall chip area is given to us we need to divide that overall chip area into some blocks which is not overlapping with each other.

So that is called rectangular dissection. So, we have a slicing floor plan. We have a slicing floor plan. What is this slicing floor plan is that will repeatedly divide your each rectangular into two smaller rectangle using a two lines called one is called the horizontal cut line other is called the vertical cut line starting with the entire chip area. So we have the complete chip area is given to us we are bisecting the total chip area into two parts to make them smaller and each of the small sub areas again is divided vertically or horizontally using the horizontal or vertical cut lines. Let us take an example here. This is the basically the vertical cut line this is V which divides the chip into two parts.

Now we have another, this is a horizontal cut line H which divides the chip area into two parts. Then we have a vertical cut line V this is a V. Now we have a horizontal cut line this is basically horizontal cut line. Now this is a vertical cut line. So, we are placing those blocks block A is placed here block B is placed here block C is placed here like that will repeat the process to place all the blocks in those sub blocks.

So, this is my this is called a slicing floor plan because it completely divides the floor plan into two different parts. So, which one is your global bounding box in this case the global bounding box is the this is your global bounding box the boundary or the perimeter of this rectangle the is called your global bounding box. The boundary or the perimeter of the of this rectangle is called your global bounding box. Now this slicing floor plan we can represent it using a slicing tree we have a binary tree with k leaves k-1 internal nodes each of the leaves is represented by a block. So, we have a binary tree which is having k leaves and k-1 internal nodes each of the leaves is basically a one of the blocks of your design and each internal nodes are basically your either the horizontal cut or vertical cut.

Your internal nodes are either the horizontal cut or the vertical cuts and the leaves are the blocks of your design. So, I will repeat it again so your leaves are the blocks and internal nodes are the horizontal or vertical cut lines. In this slide we will discuss about slicing floor plan and its corresponding slicing tree. So here we have a example of a slicing floor plan and how can we construct the slicing tree corresponding to that one. So, if you can see here this is the bounding box this one is the bounding box of the floor plan and we have a vertical cut.

So, this line is corresponding to this V node. Then we have a horizontal cut, so this is a horizontal cut which is corresponding to this one. So, after this horizontal cut we have again a vertical cut so this one is corresponding to this one. Now we have A is in the top so the right child is basically corresponding to the top block and the left child is corresponding to the bottom block. So here if you can see now, we have again a vertical cut, so the vertical cut has now again two parts which is B block and the C block. B is the left side block and B is the left side node. Similarly, the C is the right-side block and the C is the right side node. Now they are the left child B is the left child and C is the right child. Now what we have to do now this partition is over this left side is over now we will go into the right side. So, the right side we have a horizontal cut so this one is basically corresponding to this one. Now we have again a vertical cut, so this vertical cut is corresponding to this vertical cut.

Now this vertical cut has two parts one is D and E, so D is left child and E is the right child. Now we have to this top portion will go to the right side so this total thing is top of this horizontal cut so that is in the right side. So, this total is the top of the of this horizontal cut so that is that is why it is in the right side. Now we have again a horizontal cut so let us say this one is a horizontal cut, so this is corresponding to that one. Now we have two things one is basically I which is the bottom so that is a left child then we have a vertical cut basically this one so this is my vertical cut corresponding to that one.

Now again I have another vertical cut, so this is corresponding to this one. So, now we have that this vertical cut has two child one is G and H, G is left side of that so G is the

left side block so that is why it is a G is the left side node and H is in the right side so H is the right side node. So, this is the slicing tree corresponding to this slicing floor plan. So, the slicing tree can be constructed in different ways for a given floor plan. For example here if you can see this portion we have two vertical cuts so in the previous case we have used F first and this cut first and this one later and this one is corresponding to this one and this two is corresponding to this two.

However, in this slicing tree we have taken this one first, so this is corresponding to this one and this two is corresponding to this two. So, it is possible to create different slicing tree for a given slicing floor plan. So, it is possible to create different slicing tree for a given floor plan. So, in this slide we will represent the slicing tree using a polish expression. So, what is polish expression? Polish expression is basically is a way of expression representing the tree.

So here what happens is that we have a basically if you can see in this small diagram, you have a parent node where this is parent node this is the left child and this is the right child. So, in case of polish expression we denote this one by L first then R then P. So, this is the way I need to represent using the polish expression. So basically, if you can see I can write the polish expression for this slicing tree using this method.

So, what is happening here is that we have to assume two things. The V or the vertical cot is represented by star and the horizontal cot is represented by plus symbol. So now I want to represent this slicing tree using polish expression. So first we have the leaf nodes. The leaf nodes are nothing but the blocks of the floor plan and this parent nodes are nothing but your cuts either vertical cut or the horizontal cut.

So, we will represent using the polish expression. So first one is we will take the leftmost block which is the B block. Then the rightmost block is the C block. Then the type of cut is a vertical cot. The vertical cot will be represented using star BC star.

So, this is corresponding to this portion only. Then we will go to the rightmost A. Then we have basically the parent node. Parent node is plus. After this parent node then I will go to the rightmost part and in the rightmost part of the tree I will see the which is the leftmost block. Leftmost block is the I. So, this I is corresponding to this I. So, it is the right-side left child is the I and after that I will go to F. Then I will go to G.

Then I will go to H. Then we have V. The V will be represented by star. Then another V which is represented by another star. Then I have H which is represented by a plus symbol. Then I have to go to the leftmost block which is D. Then block E. Then V node which is V node will be represented by star. This V node will be represented by star. Then again, we will go to the H node which is represented by plus symbol. Then finally we will go to the V node of which is the root node which will be represented by star again. So, this is the Polish expression corresponding to this slicing tree.

So, what we learnt here? We have a slicing floor plan. From that slicing floor plan I created a slicing tree. From the slicing tree I represent that slicing tree using a Polish expression. So, the Polish expression will be easy to utilize inside the algorithms. So, this Polish expression can be used to basically create the slicing tree. From the slicing tree you can create the slicing floor plan. Then we have a non-slicing floor plan. The non-slicing floor plan is basically here we do not cut the thing completely. We do not cut the complete design area into either a vertical cut line or a horizontal cut line. So, the floor plan cannot be found by sequence of a only vertical or a horizontal cut in a parent block.

The smallest example without wasted space is your wheel. So, we will take an example of wheel in the next slides. But then we will discuss about the floor plan tree which is a reference a hierarchical floor plan. In this case your leaf nodes are your blocks of your design and your internal nodes are basically either a horizontal cut or a vertical cut or a wheel. Then we have order of the floor plan tree is the number of internal or non-leaf nodes in the floor plan tree is called the order of the floor plan.

This is an example of a floor plan. This left-hand side is your floor plan and this is the floor plan tree. This is the tree representation of the floor plan or a floor plan tree. So, we can see that here we have a wheel. The wheel we have already discussed of V and H. Wheel is important here. This portion is represented as a wheel. So, your W you have F then G then H then I and J. So that is why it is a wheel. You have a F then G then H then I and J. And rest of the things are whatever I discussed earlier vertical and horizontal cut lines.

We have basically the constraint graph. So, the constraint graph consists of edges connecting N plus to weighted nodes. We have a source node, and we have a sink node. Source nodes are denoted by S and the sink node is denoted by T or you can say termination node that is why it is T. Then you have N different blocks are there. So, the blocks are denoted by your nodes of a graph or V1, V2, VN which is representing your blocks M1, M2, Mn. So, the weight W of Vi of a node V will give you the dimension of the corresponding block. Obviously the W(S) or W(T) has dimension 0, but the other blocks will have the corresponding dimensions. So, we have this vertical constraint graph. In this case you have node weights which gives the height of the blocks. So, if you know the height of each of the blocks you can find the height of the global bounding box. So, the node weights for a vertical constraint graph helps me to find out the height of my global bounding box. And I have two let us say I have two nodes Vi and Vj which is corresponds to the block Mi and Mj. They are connected by a directed graph or a directed edge from Vi to Vj if my Mi is below Mj. So, I have a block Mi and I have a block Mj. In this case I have an edge from Mi to Mj. Let us take an example here. You have a source node whose weight is basically 0, termination node has weight 0 and we

have these edges connected them. This is called the vertical constraint graph and this will help me to find the height of my global bounding box. Similarly, I have horizontal constraint graph. Node weights are representing the width of the block.

I have two nodes let us say Vi and Vj which corresponds to the block Mi and Mj. Then if I have a direction between them, let us say I have Mi is left off, I have Mi and I have Mj. My arrow is from this to this which helps me to find the width of the global bounding box. So, this is the basically floor plan and its corresponding horizontal constraint graph. This is the left-hand side is your floor plan and right-hand side is your horizontal constraint graph. Then we have a concept called sequence pair and I have S plus and S minus of block permutations that represents the relationship between every blocks A and B.

So let us say I have A and B in S plus, I have two sequences here. Sequence pair means I have two sequences, one is S plus, and another is S minus. If I have two sequences given to me then I can tell whether my block A is left of block B or block A is above block B or block A is below block B or block A is right of block B. So, if I have a sequence AB and AB in the both sequence 1 and 2, this is let us say sequence 1 and sequence 2, then block A is left of block B. And if I have AB and BA that means block A is above block B. And if I have BA and AB then block A is below of block B. And the final one BA and BA then block A is right of block B. This sequence pair helps me to represent my floor plan in a very compact manner which can be given as input to our floor planning algorithm. So, in this lecture we discussed about floor planning and what is the difference between floor planning and placement and different terminologies involving floor planning.

Thank you for your attention. Thank you very much.