VLSI Physical Design with Timing Analysis

Dr. Bishnu Prasad Das

Department of Electronics and Communication Engineering

Indian Institute of Technology, Roorkee

Week - 08

Lecture 40

Global Routing in the connectivity graph

Welcome to the course on VLSI Physical Design with Timing Analysis. In this lecture, we will discuss how to do global routing in a connectivity graph. So, the content of this lecture includes, so how the global routing is possible using a connectivity graph, then we will discuss some algorithm, then we will discuss about the routing reasons, definition of the routing reasons, defining the connectivity graph and we will discuss some examples. So, first of all we will discuss about the global routing in a connectivity graph. So, what it does is it combines the switch boxes and channels and it handles non-rectangular block shapes. So, this connectivity graph models the switch boxes and channels and it also handles the non-linear block shapes. It is suitable for full custom design and multi-chip modules also. So, let us discuss some of the connectivity graph. So, there are couple of connectivity graph are there, we will discuss with some examples. So, let us take the first one the channel connectivity graph. This is the first one channel connectivity graph. What it says that let us say this left hand side is a layout. This is a layout and this blue rectangles are the blocks. Blocks means these are already placed. These blocks are already placed.

Let us say this is B1, this is B2, B3, B4, B5 and B6. So, these blocks are already placed. So, if these blocks are placed, what are the void spaces are there. So, those are called the channels. So, we can define this whole region from here to here as channel 1 channel. That will represent one node in a graph. So, this portion will represent as a node in a graph. So, whenever we are defining a graph, we have two things. One is vertex, the graph G is defined by V, E. So, we have a vertex set, we have an edge set.

So, this vertex sets are basically in this case will be basically the channels. So, the first channel is one here, whatever I showed here. So, then similar concept will be applicable for all the channels and will create the vertices of the graph. Now then we will define the edges. How the edges will look like? The edges are the channels which are basically adjacent or connected. So, let us say if you can see the first one, the first node or the first channel is connected to 2, 4 and 8. So, if you can see the graph from 1 to 2, we have an edge which corresponds to this one. The 1 to 2, there is a 1 and the 2 channels are neighbors. Similarly if you can see the 1 and 4, they are neighbors and 1 and 8 are the neighbors. So, there is an edge here.

So, this edge corresponds to this one. So, similarly we will create the edges for all the neighboring nodes or the vertices. Now we have a concept called switch box connectivity graph. So, this is the second type of connectivity graph, the switch box connectivity graph. What it does is that we have the crossing points.

Crossing points means let us say you have a vertical channel and a horizontal channel. So, if you can see here in the previous diagram, this is a vertical channel, the node 1, this one is a vertical channel and this 4 is a horizontal channel. When a vertical channel crosses a horizontal channel, then there creates a switch box. So, if you can see the next diagram, the 1 and the 4 channel in the connectivity graphs are basically crossed at what are the points. So, here if you can see here there is a switch box which corresponds to this one. Similarly 1 and the 4 and the 3 is connected here. So, this creates a switch box. Similarly we have horizontal channel 4 and the vertical channel 9 are crossed at this point and it creates a node 11 in case of a switch box connectivity graph. So, this one corresponds to this one. So, in this manner, we will create the switch box connectivity graph. So, wherever we have a horizontal channel and a vertical channel meets each other, there it will create a switch box and that will correspond to the vertices of that graph. So, this graph G is V, E, this vertex set corresponds to the switch boxes and the edge set is basically the neighbouring switch boxes or basically adjacent switch boxes. So, the 1 and 4 are neighbours. So, there is an edge between them. This edge is because corresponds to this edge. Similarly this edge corresponds to this edge. So, it is very easy to draw switch box connectivity graph. Now we have basically channel connectivity graph. Now we have switch box connectivity graph. We can combine both of them to create a connectivity graph.

So, if you can see here, basically whatever I told you these are the switch boxes. This corresponds to your blue nodes, this corresponds to your blue nodes, this will corresponds to this one and if you can see this one will corresponds to this one like this. Now we have some channel in between. So, channels are connected with another colour. So, this one will be the channel between the switch boxes. So, this will corresponds to the channel between the switch boxes. Now we will discuss about an algorithm how we can do global routing in a connectivity graph. So, in this case what is given to us as input to the algorithm. So, we have provided with two things. First one is the layout.

The layout of the design is given to us and the netlist, how the nets are connected that is also given to us. These two are given to us. One is the netlist, how the nets are connected, how the pins are connected and the second one is the layout of the design, layout of the overall design that is given to us. Then what is our problem statement here? Our problem statement is how we can route the net such that we have minimum wire length we are used in the routing. So, we have to define a routing scheme for each net in the netlist. We have to define a routing approach such that each net in the netlist will be routed in a shortest path. So, our objective is to route the net with shortest path. So, how we can do this? It involves some of the steps. The first step is called defining the routing regions. The first step is basically defining the routing regions.

So, what is the input to that function? Input to that function is our layout la. So, layout la is input to our function define routing region. So, the output is rr, so that will go as input to the second function which will create or define the connectivity graph. So, this rr will be input to our connectivity graph function or define connectivity graph function and it will return cg. It will create a connectivity graph of the layout.

Finally, the first two steps will create a connectivity graph of the layout. Then the third step is basically net ordering. So, what is this net ordering? Basically you have the second parameter, the first parameter whatever given here that will be input to that algorithm, to that function. What is that function? Net ordering. So, we are giving the netlist to that function and what it will determine that in order at which the nets will be routed, the order at which the nets will be routed and it will define also the priority of each of the nets.

Then after you do the net ordering because some of the nets will be critical, so the critical nets will have the highest priority. So, now after the net ordering is done what we have to provide is that we have this rr routing region and the netlist provided to the function assign Tracks. So, we will assign tracks for all the pins connected in the netlist. So, we have the routing regions is given to us, we have the netlist given to us, then the then it will assign track to those pins in the netlist, all the pins in the netlist. Then what we have to do is that we consider one net at a time.

We will consider one net at a time. So, we will take first net, then it will basically pass through a free track. If the track is free then that net will be assigned to that track. Then it will decompose the net into two pin subnets actually. It will break the nets into two pin subnets. Then this function subnet will create the subnet of the given net. Subnet has multiple lines, it will create a subnet of that net. Then each of the subnet, so there is a loop j equals 1 to subnet for a given net will create the subnet assign them independently. So, this first subnet is taken into account. Then what we have to do is that we have to have this S path which finds the shortest path from the connectivity graph.

From the connectivity graph it will find the shortest path for that subnet. So, if no shortest path exists it will not route. Otherwise the nodes on the shortest path will be assigned the track and basically the routing capacity of that track will be updated. Since one net is assigned to that track then the other net should not go through that track. So, we need to redefine or update the routing capacity of all the tracks in the connectivity graph. So, this is all about the pseudo code of the algorithm but we will discuss step by step how things are happening with an example. Now we will discuss this global routing using in a connectivity graph stepwise. The step one basically defines the routing reasons. So, how we can define the routing reason in this case? We have basically vertical and horizontal reasons are formed. So, the first point if you go to this algorithm the first point is the defined routing reasons of this first point.

First step, so what is happening here? What is input to us? The layout. So, these are the blocks actually. This is block 1, let us say this is block 2, let us say this is block 3. So, these block are the places where we cannot do any kind of routing. So, these are the basically places where our actual design is sitting. We need to connect those nets to do the complete design. So, what the first step is we have to create the vertical and horizontal routing reasons. So, this is the layout is input and the output is basically the vertical routing reasons and the horizontal routing reasons. So, if I can show you a horizontal routing reason this is a horizontal routing reason. So, this is a, now this one is a vertical routing reason. This one is a vertical routing reason. Now then second step is stretch the bounding box of the cell, cell means the blocks, bounding box of the cell in each direction until a cell or a chip boundary is reached. So, what it does, let us say I have this is a cell boundary. Let us say this is the block or the cell whatever you can say. Let us say this is let us say B1. So, I will have basically these lines are basically touching your bounding box. It is touching your bounding box till the chip boundary. The chip boundary is this one. This is chip boundary. Now either it will touch the chip boundary or a second cell. The second cell is the B3 here. So, it will stop here. So, we are creating basically channels here. We are indirectly creating channels here. Now what is happened is that we have created the routing reasons. We have created the routing reasons. Then we are now creating and defining the connectivity graph. Now the second step is defining the connectivity graph. So, if you can see here we have basically this light hand side is the layout. This is the layout which is broken down into the routing reasons. Now from the routing reasons we will define the connectivity graph.

So, what is the concept here is that whenever I have a graph I have vertices or the nodes we have edges. So, for each of the routing reasons we have vertices. So, let us for example this one, this is the vertices. For this one, this is the vertices. So, the nodes are representing your routing reasons. From layout we are broken down into the routing reasons and from routing reasons now we are defining the connectivity graph. In the connectivity graph I have vertices and edges. So, I have already defined the vertices. Now edges. So, edges is what the connection between the routing reasons.

If two routing reasons is adjacent to each other then there is an edge between them. For example here from 1 to 8 the 1 and 8 are adjacent. The 1 and 8 are adjacent. So, if you can go here this edge the adjacency is represented by this edge here. So, similarly you can draw all the edges of the connectivity graph. Then one second point is that your edge contains two things. One is horizontal capacity and vertical capacity of the routing reasons. So, what is this capacity says? How many interconnect can pass through that reasons? How many wires or metals can pass through that routing reasons? So, here if you can see here for each of the nodes we define that. So, if you have this is the routing region 1. So, this basically horizontal and vertical capacities are stored in the routing regions.

It is denoted in the nodes of the connectivity graph. Nodes contain this horizontal and vertical capacity. So, if you can see here in the node 1 we have two basically horizontal tracks and one vertical tracks are there. So, the horizontal tracks is denoted by first so here it is 1. So, this is horizontal tracks this one corresponds to the horizontal capacity and this one corresponds to the vertical capacity. So, for the routing region 1 we have one horizontal track and two vertical tracks.

Similarly you can define it for all the regions. Similarly all the routing regions. Similarly this is the connectivity graph for this layout. Now after we created the connectivity graph we have some steps involved. So, one of the step is net ordering. Determine the order of the nets by which it will be routed. So, nets can be given priority based on criticality of the path or the interconnect. Then it depends upon number of pins it connects and it also depends upon the size of the bounding box. If you have a size of the bounding box is large then it will get higher priority. And some of the algorithms also update the priority dynamically. So, some of the algorithms can update the priority dynamically. So, the step 4 basically assigns the tracks for all the pin connections. So, for a given pin a horizontal track and a vertical track are preserved within the pins routing regions. So, let us say if I have a pin in the design or a layout so a horizontal track and a vertical tracks are reserved for that pin. So, this is called assigning the tracks to the pin. Then we have a routing global routing of all the nets. So, what we did is that we have nets and each net has subnets. So, we do nets and subnet ordering the order at which they can be routed. Then we do track assignment in the connectivity graph. We will discuss this algorithm called mesh routing. Then we will update the capacity in a connectivity graph. So, now we will discuss one example. In this example what are given to us is that we have a layout. The first one whatever I told is the layout. So, this is the layout of our design.

Now the second thing is given is your nets A and B. Then from that here the basically connectivity graph is given but we can create the connectivity graph also from the layout itself. We can create the connectivity graph from the layout itself. Then what is our basically objective is how we can route these two nets A and B using less interconnects, less fewer resources as possible. So, this is the connectivity graph corresponding to this layout.

Let us take the pin A. So, if I use the pin A then the track one of the track will be used. One of the track will be used. One of the horizontal track, one of the vertical tracks will be used. So, earlier in this routing region whatever the routing resources was there that routing resources will be reduced by 1. If I can go to the previous diagram, so earlier the routing resources was there 4, 2 for this region 2. So, the 4, 2 will be reduced by 1 in the vertical direction, 1 in the horizontal direction if I do that routing. The final value after you do the A connection then it will come to 3, 1. So, that is the reason this becomes 3, 1. Now we have two reasons, two nodes we are discovering.

We are going through two nodes. What are those two nodes and what is the resources before there? So, here the resources before routing is 1 and 2. Here the resources before routing is 1 and 5. Now after you do the routing, this node capacity will be reduced to 0, 1 and this node capacity will be reduced to 0, 4. So, if you can go and check, correct. So, 0, 1 and 0, 4. So, now the net A is routed. So, what happen is that here we are given priority to the net A. So, that is why the net A is routed first. Now the net B will be routed. So, whenever the net B will be routed, so there is one more point here I want to tell you that here we have two metals used. This one is let us say horizontal metal, this is horizontal metal and this one is a vertical metal. Let us say if I use two metal, it can be M1, it can be M2. So, now we will consider the Bnet. So, whenever I am doing the Bnet, I need to similarly check for how my routing resources will change in node 2, node 3, node 7, node 12, node 11, node 10 and finally node 8. So, if you can see here, wherever I am using both the metals, both the horizontal and vertical metals, in that case both the routing resources will be reduced by 1. But for example, if you can see in this node 11, we are just using the horizontal metal. So, vertical metal will not change. If you can go here, this one is 1, 7 here after routing.

So, the before routing condition at this point should be 2, 7. Let us go and check. So, this is 2, 7. So, similarly here it is 4, 2 before routing. Then after routing, it should be basically since vertical metal is used, it should come to, it will be 4, 1. So, wherever I use both the metals, so both the horizontal and vertical capacity will be reduced by 1. If I use the horizontal metal, then the horizontal capacity will be reduced by 1, vertical capacity will not change and vice versa.

So, this is a finally routed design. So, we have another example. First one is the layout. This is the layout given to us. This is the layout of the design given to us. Then we have nets. Nets are A. These two nets should be connected. Now B. These two nets should be connected. So, then if these nets and the layouts are given to us, we can create the connectivity graph. So, the right hand side shows a connectivity graph. So, our aim is to basically find the routability of both the nets A and B. So, we are considering the net A first. For routing the net A in a shortest path, we are using the routing regions 4, 5, 6, 7 and 10. So, if I use that net, so basically my routing regions basically will be updated from 0 and 4 will be 0, 3.

This will be updated from 1, 1 because I am using a horizontal routing track here. This corresponds to the horizontal track. So, the horizontal value is updated and here it is 0, 3 because I am using a vertical track here. So, this is a vertical track, vertical metal. So, 0, 3. Then in case of 5 it is 0, 1 because only horizontal track is used. Here also 0, 4 because of horizontal track is used. Here it will be 0, 2 because I am using horizontal and vertical also. Here there is a via will be there between metal horizontal and vertical metal. Now this will be again reduced by one in the vertical direction, one in the horizontal direction. Now after I do this pin A routing, the pin B there is no space in the routing region 4, 5, 6 for the net B. So, for doing the routing of the net B, we cannot route in the shortest path. What we have to do is that we have to go through the nodes 4. For net B, we need to go through the routing regions 4, then 8, then 9, then 1. So, this is 1, then 2, then 6. So, we cannot route the net B in a shortest path because of the basically routing condition and we use a ditter to do this routing, but the placement is routable.

Finally the placement is routable. So, this is the final routing or the global routing in the final layout for the net A and B.

Thank you for your attention.