VLSI Physical Design with Timing Analysis

Dr. Bishnu Prasad Das

Department of Electronics and Communication Engineering

Indian Institute of Technology Roorkee

Week 09

Lecture 44

Channel Routing Algorithms – I

Welcome to the course on VLSI Physical Design with Timing Analysis. In this lecture, we will discuss about channel routing algorithms. So, the content of this lecture includes first of all we will discuss about the vertical constraint graph, then we will discuss channel routing algorithm, we will discuss left edge algorithm. In the last class, we discussed about the horizontal constraint graph. For any channel, we also need to create a vertical constraint graph. So, in this lecture, we will start with the vertical constraint graph whatever we left in the last lecture. So, what is vertical constraint graph? It is any graph is denoted by two things, one is called basically nodes and edges. So, nodes are basically the pins of the pins in the channel, nodes are basically the pins in the channel and the edge directed edge e (i, j) belongs to E connects the node i and j if the horizontal segment of node i must be located above the node j. For example, if you can see here, you have a net A in the top side and net B in the bottom side. So, there is edge from node A to node j in the vertical constraint graph.

So, the net basically the direction should start basically the arrow should start from the top pin and it ends with the bottom pin. So, here the A pin corresponds to this node and the B pin corresponds to this node. Now we will discuss about one more concept called transitivity. What is this transitivity? Transitivity says that and is that can be derived by transitivity is not included. So, let us here if you can see you have basically let us say I will consider the nodes B here, node for example, here if you can see this segment, if you consider this segment, I have a B node here and I have a C node here.

So, that should be edge is from B to C. Then if you can see here, then you have D node here, then you have E node here. So, there is edge is from D to E. This is corresponding to this one. Now you have E to C is also there.

Then the next one, if I consider the next segment, here you have a E node and C node. Now if you can see that, so D is connected to E and E is connected to C. Then if you can see another segment B, then B node is connected to E node. We can remove the redundant edges. If you can see here from B node, we have a B node and D node is there.

We have edge to E node and there is a edge from E node to C node. So, if you can see here, B is connected to E. So, this line corresponds to this one and basically this line, I will put a star here and this line corresponds to this node star here. These two are the same edges. Similarly, from E to C, you can let us say I will put this edge corresponds to this edge here.

Now, B to C edge is not needed. So, B to C edge is not needed. So, this we can remove from the graph and edge that can be derived by transitivity is not included. For example, the B to C node is not included in the vertical constraint graph because there is an intermediate node which is there, which is connecting both of them. So, now there is a concept called cycle.

So, what is the cycle in a vertical constraint graph? So, the cycle in a vertical constraint graph indicates a conflict where there is a vertical segment of the two nets overlapped at a specific column. So, this cycle basically creates a conflict of the vertical segment. Let us say if I have two nets, I have net A and net B, I have an edge from net A to net B, this edge. And so, if I go to this one B and A here, I have a net corresponding to this edge from B to A. So, this creates a cycle.

So, in this case, we cannot use the same vertical segment for routing in the same column, routing the A and B net in the same column. So, this problem can be solved using net splitting. So, what we did here, we have introduced the net. So, the B1 net divided the B1 net into two parts. So, here you have a net from here to here. Let us say this you call this one B1 and from here to here, there is a net called B2. So, introducing this net splitting concept, we can do the routing when there is a cycle. We will discuss this one in more detail in this lecture, end of this lecture.

So, now, we have this channel is there, how we can find the vertical constraint graph and the horizontal constraint graph for the nets A to F. So, we have nets A to F is there, how can I create HCG and VCG. So, we have basically two sets there, which is basically giving the net IDs. So, we have basically columns, these are the columns and these are the nets. So, the top nets and these are the bottom nets. So, these are the net IDs for the top, we have 0, this corresponds to this 0, this B corresponds to this B and this E corresponds to this E. Similarly, for the bottom, this D corresponds to this D and this F corresponds to this F.

Now what we have to find out, what are the basically S of the column. Now we are finding S of the column. The S of the column is basically the nets going through that column. So, let us say here we have this column A, for corresponding to A, what are the nets going in the column A, only the D net, this D is corresponding to this D. Now, for S of B, if you can see S of B, this is your B, this S of B, we have this, I will write it in a different colour. So, S of B corresponds to B, B should be there because this is the B column and C should be there definitely, B C is corresponding to this B C and D should be there. Why D should be there? Because this D net will pass through this column. So, this D net should pass through this column to connect with this B net. So, D will also be included when we are calculating S of B. Then we can calculate S of C, S of D like that for all the columns.

So, if you can see, if there is any kind of subset of S there that should be removed. If you can see here, let us say consider this first three, the first three I will consider. Let us say this is this first one, this is the second one and this is the third one. So, S of A and S of B are subset of S of C. So, D is there here and D is also there.

So, I do not need to represent S of A. Similarly, B C D of S of B is also there in S of C. So, S of A and S of B should not be considered. So, S of A and S of B are both subset of S of C. So, this A and S of B should not be considered.

Similarly, S of F and S of G are both subset of S of E. So, this basically S of A, S of B, S of F, S of G are not considered. So, are not considered for HCG because these are related to horizontal constraint graph. Now, what we are doing is that we are creating a basically diagram here which is only depend upon basically the S of C, S of E and S of D. So, these three, so S of C, S of D and S of E.

So, if you can see here S of C and S of D and S of E is there. So, what this B says that B is there in both the columns and C is there in all the, B is there in both the columns and C is there in all the columns. Then the D is there here and F is there in both the columns, then E is there in all the columns. So, now what we are doing is that we are drawing the HCG from the left side. So, here whenever we are creating the HCG we have to refer to this diagram.

So, here basically what we are doing for B node, let us say this B node should be connected to C, D, E. So, I will consider first the B node. So, the B node should be connected to this 1, 2, 3, C, D, E. This is C, this is D, this is E. Now after these connections then you can see the C should be connected to D and E.

So, now I can use another colors, thus you have the now we are concentrating on the C node which is connected to D, this D node and connected to E node. We have edges already, this edge and this edge. Similarly, D should be connected to your E node. See D is already connected to E node. So, this same approach will be applicable for all the columns.

In this manner we will create the HCG. Now we will consider the VCG vertical constraint graph. Whatever the way I just now discussed, we will take a pin from any one of the top pins, then what is the pins below it, there will be edge between them. And we can also remove the edges if there is any kind of redundant edges. So, what we discussed, same manner we can create this VCG.

So, if you can go here, if you can go here the B should be connected to C. B is connected to C, we have a edge. Similarly D and E, one edge is there. So, this method will repeat, but any redundant edges should also be removed. So, this is the vertical constraint graph. Since there is no cycle in the vertical constraint graph, so no net splitting is required and thus each net only needs only one horizontal track for routing. So, one important point is that whenever you have a cycle, you need an extra track. Here if you can go back, whenever there is a cycle is there, in this case you need one extra track. So, number of track here is basically this is the first track corresponding to this line, this is the second track corresponding to this line, this is the third track corresponding to this line. So, here you need an extra track for basically because of the net splitting.

So, now since there are no cycle and conflict in the VCG, the minimum number of tracks is equal to the cardinality of the largest S column. So, if you can see here the largest S column is S of C and S of E, then the cardinality of that one is 4. So, this can be routed with 4 horizontal routing tracks.

So, now we will discuss about the channel routing algorithm. So, here the main objective is to minimize the number of tracks. So, minimize the number of tracks. So, there are two popular algorithms are there, two popular algorithms are there. One is the left edge algorithm and the second is called the dogleg routing algorithm. So, first of all we will discuss about this left edge algorithm. It is based on the vertical constraint graph and the zone representation and it is greedily maximize the usage of the track. So, here two things are needed, one is basically the first one is basically vertical constraint graph, the way we discussed in the last example. So, this vertical constraint graph is essential for assigning the order of the nets to the track. So, vertical constraint graph is useful for assigning the order at which the nets will be assigned to the tracks. So, we have multiple nets which nets should be has more priority that will determine from the vertical constraint graph. Then the second one is the zone representation which determines which net may share the same track. So, here if there is a conflict, so then the same track will not be assigned. If there is no conflict, then the nets can share the track. So, there is one point if no conflict, then share the track. Then if conflict, then sharing is not possible, sharing of track is not possible. So, this is the most important part of this left edge algorithm, and each net uses only one horizontal track that is called the trunk. Now we will discuss this left edge algorithm pseudo code.

So, then we will discuss an example. So, basically if you can see here, then what is the input to your channel routing basically algorithm, the left edge algorithm is your channel instance, routing channel instance that CR, channel routing instance that is CR, then what is the output, how this each of the nets have been assigned to the track, basically the track assignment of all the nets. So, the first we will start with the topmost track actually, the topmost track which is the current track. So, we will start with the current track which is 1. Then we will assign all the netlist, basically nets are not assigned. So, it is an initialization step. So, all the netlist should be net on assigned. So, all the nets are on assigned in the second step, this is the initialization step. So, basically, we have all the nets are on assigned at the first entering the while loop and one of the nets will be taken at a time. So, then what is happening is that we are creating the basically vertical constraint graph and zones. So, if you can this the first one is a function which creates a vertical constraint graph of the routing instance or the channel routing instance CR.

So, input is basically the CR and output is your vertical constraint graph. Then you have what is happening is that you have a zone representation. So, we have zone representation happens in the fifth line. So, then after you do this we will assign the nets from the left to right order. We will assign the basically the nets to the track from left to right order.

That is the reason it is called left edge algorithm. So, it will start the assigning the nets to the track from the left side. So, you have what is going inside is the net on assigned and the start column. Start column means that the first column let us say the in this example the first column it will start from the left side. So, you have basically all the nets and the start column is given then it will take the left side net and do the assignments.

So, for i equal to 1 to the all the on assigned nets. So, now your current net is basically the first net which is there in the net list and here it checks two things. One is if the current net has no parents and it does not have any conflict on the track these two are the same if loop. So, these are two conditions. The one condition is that if your current net has no parents and if it does not has conflict on current track. So, then what you have to do you have to assign that net to the current track.

There are two things there should not be any parent and there should not be any conflict. Then you can assign that net to the current track. Then once you assign that net to the current track then you your this net has already assigned in the track then you remove that net from the on assigned set of the nets. So, you have a set of nets which is on assigned from that the current net you remove that. Then what you have to do is that you increase the track increase the track to track plus 1.

You are now we are going to the next track to assign the nets. We will come back to this algorithm again after discussing one example. Here what is happening is that we are we have a left edge algorithm to route the nets A to J in this channel. So, this channel is given the first step is basically generate the VCG vertical constraint graph and the zones. So, give you some example how this graph is drawn for continuity but I will not draw the graph completely.

For example, if you can see you have a edge from A to C and also E to C. You can see here A to C and E to C is there. So, what does it mean that A to C you have a edge then E to C you have a edge. So, what it is doing that A to C edge is not used because of there is a edge from A to E and E to C. So, if you can see you have A node then you have a E node and C node is there. So, this A to C edge should not be considered. So, you have edge from A to C and E to C. Similarly, you have A to E. If you can see this one you have a edge from A to E. So, these three edges can be reduced to two edges like A then E then C. So, what is there in the graph? So, now what we are doing is that we have zone representation.

So, we have A is having this zone then the G is having this zone like that these are the zone representation. Now we consider a next track. Consider a track then find the left to right ordering of all the unassigned nodes. We will find the left to right ordering of all unassigned nodes. So, we have two conditions whatever I told in the pseudo code. We are considering the current net has no parents. So, basically those are having the parents we are not considering. So, we are giving priority to the parent node. Parent node which one is the parent node here? A is the parent node for E node.

So, this is the parent node and the E is the child node. So, if a node has no parent then those nodes will be considered. So, here A has no parent so A node is considered and D node is considered and J node is considered. But if you can see here there are two things. This is the first condition and the second condition is that there should not be any conflict.

So, if you can see here A and D has a conflict. If you can see here A and D has a conflict. So, we cannot assign A and D in the same track. But A and J has no conflict because A is running till this point and your J is running in the other extreme conditions. So, A and J can be assigned the same track. So, after you assign A and J to the track 1 then you can delete those nets from the vertical constraint graph and zone representation.

Now I have the updated VCG. Now I will do the next node which is having no parents that is the D node. And D node has no parents and it does not have any conflict on the current track and assign the second track to the net D. Net D will have the second track but you cannot consider any other node like G because there is a conflict with D to G. So, what is happening here the D will be assigned the track 2 and after you assign the net D to the track 2 will delete that node B from the vertical constraint graph and the zone representations.

Now after the D has been assigned we will look into the next node E and G. Here in this case we will consider nodes E and I and G all the three nodes I will consider. So, this is a very interesting point here. We have E, I and G has no parent this is the first conditions no parent. And second thing is that I and G has conflict.

And we have basically which we have a priority of the left edge. So, which is the left edge? E is the left edge. E is the net which is having the left edge. So, we will first assign E. So, after the E is assigned track 3 now, we have to choose either I and G. But if you can see which is the left of I and G. Left of I and G is the G. So, G is coming first in the left side and the I is coming later. So, we can assign E net after that we can assign the G net in the track 3. So, there is no conflict. Now we will consider the next track. So, what are the nodes which has no parents? So, in this case the node C, I and F has no parents.

So, if you can see here C, I, F has no parents but and is there any conflict among them? So, if you can see here they do not have any conflict among them so they can be assigned the same track. So, this is the track number 4. We have C, then we have F, then we have I.

This is the track 4. Now we have basically left with H and B. And these are the last two nodes they do not have parents H and B has no parents. And if you can see here they do not have conflict so the track 5 will be assigned to B because B is the left edge then H. So, now we have finally assigned the nets to the track. So, this is a final basically routing. So, if you can see this nets how it is routed without any conflict. So, this is the beauty of this left edge algorithm. It starts from the left edge and do the routing with less number of tracks and it also shares the nets which do not have any kind of conflict.

So, in this lecture we discussed about vertical constraint graph and we discussed about one of the channel routing algorithm that is the most popular left edge algorithm. We discussed the pseudo code and with an example.

Thank you for your attention. Thank you.