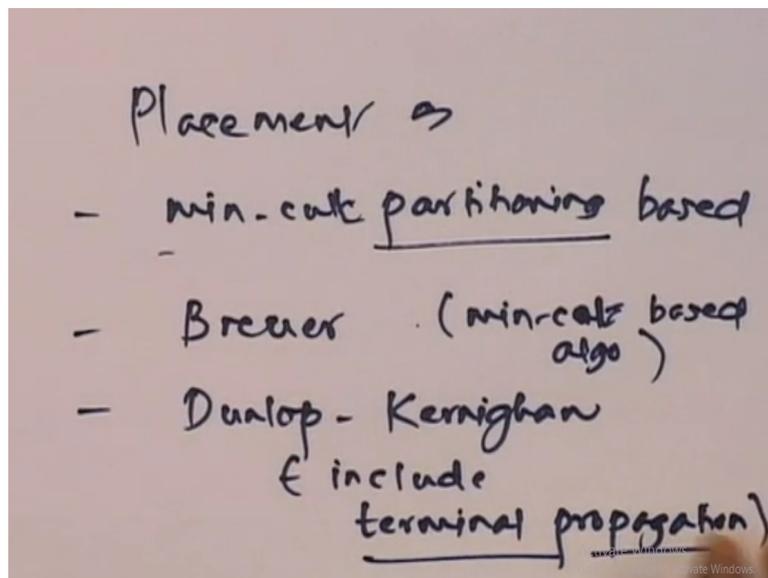


**Advanced VLSI Design**  
**Prof. Sachin Patkar**  
**Department of Electrical Engineering**  
**Indian Institute of Technology – Bombay**

**Lecture - 32**  
**Placement algorithm**

Welcome, so in this hopefully a short lecture, shorter lecture I want to give an outline or illustration of an interesting like placement algorithm for placement that gives us the idea of min cut partitioning.

**(Refer Slide Time: 00:36)**



Okay, so through an example I will illustrate some of the important role of partitioning in design of a placement algorithm. In fact, like as I described in the last lecture, in the previous lecture there are different kinds of placement. Placement is very important phase of physical design automation and there are various different kinds of algorithms for placement. In fact, a very rich area lot of algorithms have been, ideas have been researched upon.

And broadly one can classify these placement algorithms as like recursive which are top down which are based on this kind of partitioning idea. Then there are some algorithms which are iterative which are based on with a celebrated, simulated annealing idea or like this physics based idea of mass springs called forced directed placement. There are some very interesting mathematical linear algebraic approaches, which make use of the Argon spectrum and of connectivity information.

There is a matrix called Laplacian matrix which is defined using the connectivity information. It is Argon spectrum that is Argon values and Argon vectors are used to like solve a geometric version of this partitioning problem and from that combinational solution for partitioning is cooked up.

So very interesting different types of placement algorithms have been invented are being used in the CAT tools, so for the limited scope of this lectures as an introduction or overview of some physical design concepts, I would just focus on the partitioning based recursive or top down algorithm. Again very natural divide conquer idea and just want to connect it up with the important role of partitioning.

I mean a partitioning can be done manually at the level the designer has kind of indirectly implicitly specified. If designer has specified there is a datapath, there is a controller obviously natural partition would respect that the layout for a controller should be separately done layout for a datapath should be separately done. If datapath further is kind of like logically built up using some portions like fetch unit, decode unit, execute unit.

Then the partitioning can manual partitioning or a designer specific partitioning, implied partitioning could like induce the placement algorithm to look at the separate logical portions like fetch unit, decode unit, in case of a CPU datapath to be implemented to be processed for the layout generation separately and then things can be stitched together. That is a divide conquer approach.

And many of the times the partitioning is very natural is kind of implicit in the design itself or can be hinted or specified by the designer in a manual way. On the other hand, if there is no clue about if we are looking at a subsystem and we want to make a good layout of it. We already have a floor plan, we have identified a part of the chip to be used for the layout of sub systems, sub design.

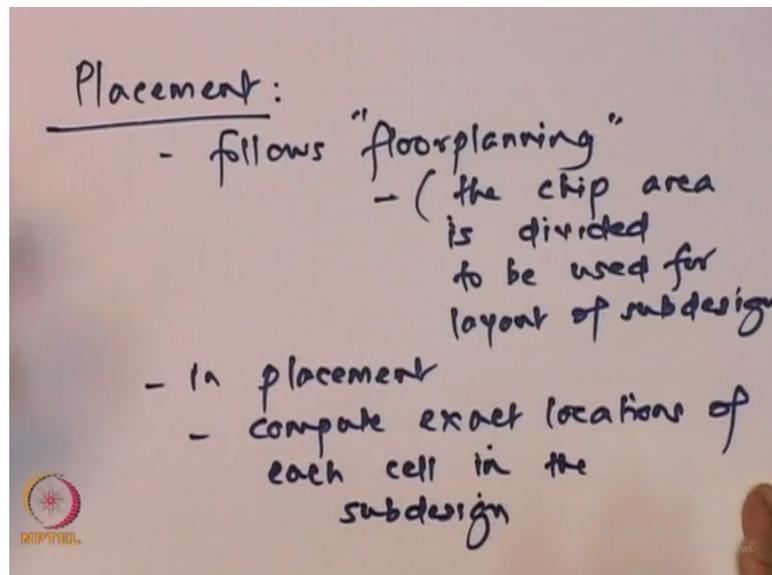
And that sub design is still complex enough that we cannot get a placement or a routing in an ad hoc fashion very easily. So we want to further reduce the complexity of this sub design by partitioning and corresponding area of the chip to be partitioned and this parts block of this sub design could be separately laid out on the parts of the chip area which is meant for this sub design and so on and so forth.

So that is the notions behind that I want illustrate one example in this, in this particular lecture, okay? So it is going to be based on min cut partitioning approach. It is, basically is based, this discussion is based on an algorithm of Breuer and then an extension of that by Dunlop and Kernighans, okay. Breuer this is actually like the min cut idea the top down idea was use Breuer like give a clear definition of our min cut based algorithm for placement.

And Dunlop Kernighan extended it to include the so called terminal propagation idea. Which is a very neat like nice idea and without that one would actually not get a good like solutions even though the partitioning can be powerful, the request paradigm is obviously entered. But the terminal approach is extremely important and helps in the quality of solutions as we will see in this example, terminal propagation I mean.

Propagation of terminal, terminal meaning like input output pads that information is going to be propagated as we go recursively down in this process of min cut based drop down partition.

**(Refer Slide Time: 06:23)**



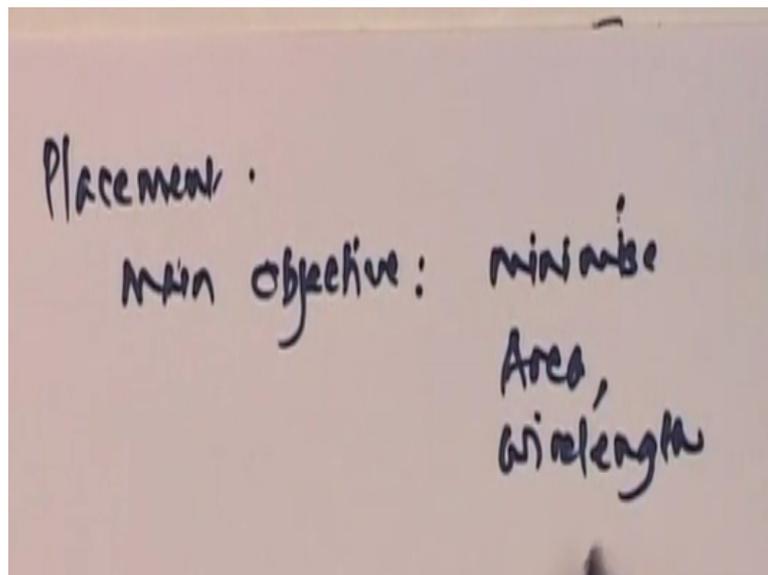
It may recall certain definitions motions again. Placement it follows floor plan, floor planning stage. In the floor planning stage here chip area is divided of sub designs, sub circuits, okay. So floor planning is one of the earlier stages which is where some kind of logical partitioning, manual partitioning would be used and corresponding the chip area portions of chip area would be identified and they have to be earmarked for the layout of sub designs, okay.

And specially in placement we compute or we figure out exact locations of each cell in the sub design, okay. So we are already working on a part of a circuit and part of chip and within that we want to like identify the exact locations where this cells of that particular sub-design should be placed. Now for the moment like for this lecture it will help to kind of forget about the floor planning parts. So we just assume that we had the full design.

And there is no floor planning the whole entire area is chip area is available and you want to divide the chip, we want to use this min cut based approach to divide a chip and correspondingly divide the circuit and independently try and like not too independently but as far as possible in a decoupled way, lay out the sub circuits on to the portions of the chip and then stitch things together in an effective way.

So we might as well ignore the floor planning issue and we just assume that we have the full design is being placed on a given chip area.

**(Refer Slide Time: 09:10)**



One of the main objectives of placement is minimize area and wire length, total wire length or maximum wire length and so on and so forth, okay total or maximum. I mean you would wonder why total is important. Of course, total is important routing but in terms of delay maximum wire length is more important but like both kind of influence each other's, so it is very difficult to get very clean optimization objective formulations in this area.

So lot of intuition domain knowledge has to be used to and lot of benchmarking has to be done to kind of validate the algorithm idea is proposed. The core algorithm development can

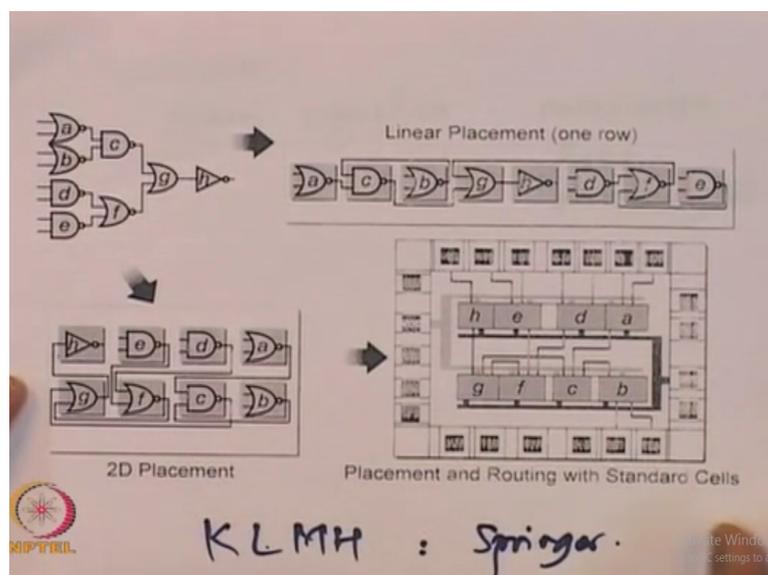
be on an abstracted problem which may not actually reflect the reality fully thoroughly but because of this sophistication of the ideas one hopes that appropriate and you offer lot of scope for appropriate modification of those ideas too like help.

So, in placement main objective of placement algorithm is to minimize area and the wire length. Wire length one can question whether the total wire length is to be minimized or maximum wire length is to be minimized. Now that is not to clear of course total wire length is going to influence the complexity that a router will have to work with and on the other hand the maximum wire length is the main like could be more important from the delay perspective, okay.

But neither of this is going to be a clean like optimization objective formulation. So in fact it is typically hard to like formulate like very correct accurate objective formulations and for those the formulations can be so complex that they would not, one would not be able to make use of interesting sophisticated mathematical or algorithmic ideas for that so, lot of emphasis is placed on getting good abstractions.

And then like modifying doing the post processing, or like modifying those ideas in a slightly juristic way for to reflect the practical considerations, practical objectives. Okay, so will not worry too much about that, we will roughly say that wire length is to be optimized, it would mean the total wire length it could partly mean that maximum wire length is to be minimized, it could also partly mean that congestion of wires is like is to be minimized whatever.

**(Refer Slide Time: 12:04)**



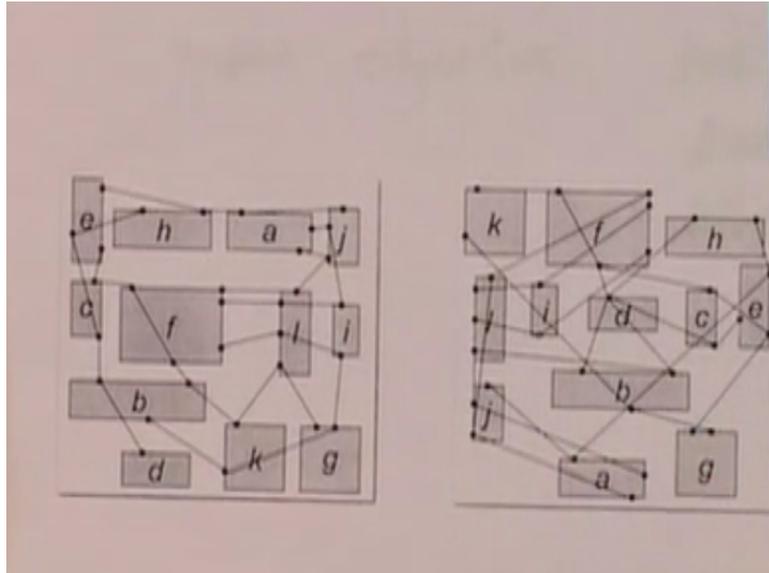
So here is, hopefully this is an example of a circuit. Again this is from the book like by Kahng, Lienig, Markov and Hu, Springer, the book title physical design from graph partition to timing closure, yeah. So this is the net list, logical net list and let say, yeah I mean the objective is of the placement is to could be to create a layout for one dimensional layout or for a 2 dimensional layout and then followed by router, which will also complete a routing and connecting up to this I/O pads and give you a layout of the chip.

Symbolically, this is how a one dimensional layout look like, this is how 2 dimensional placements will look like. One dimensional placement two dimensional placement. This might look a bit funny but yes, I mean a lot of problems can be abstracted to one dimensional placement where you have a row of standard cells to keep large number of gates on to that and then you can go to the something similar with the other rows.

So which is like practical useful but more typically we are going to have 2 dimensional layout issues, okay. Problems to be solved. So this circuit like which has this 8 gates is to be laid out on a 2 dimensional placement or 1 dimensional placement and depending on how we choose the locations of this A, B, C, D, E, F, G, H whatever, it would give us different wire length estimates so different kind of wiring complexity.

So this routing can become difficult if this particular placement where far worse. It is not clear whether this is the best placement. Anyway this is just an example how things would look like what is the objective of this like placement algorithms. So here is another example again it is not the details are not too important.

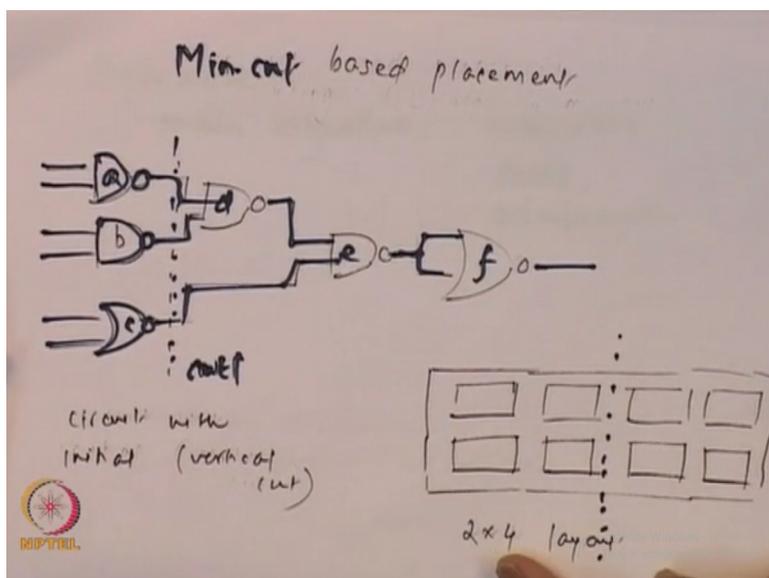
**(Refer Slide Time: 14:14)**



So on the left you see one instance of a placement of the same site, not same some set of cells A to whatever and there are some wires which are connecting them. If they are placed in this kind of positions, then the wires into run in some may be disorganized but not too congested way. On the other hand, for the same set of like cells if you make a different placement then the same set of wires logical connections.

They will look far more consistent when they are laid out. This is not actual lay out or routing of the wires. But even if it abstractly this is going to indicated this more scope, I mean there is more possibility of a bigger congestion and more difficult in routing and the quality of the lay out, okay. So the placement decisions would matter like a lot in getting a good quality lay out, okay?

**(Refer Slide Time: 15:14)**



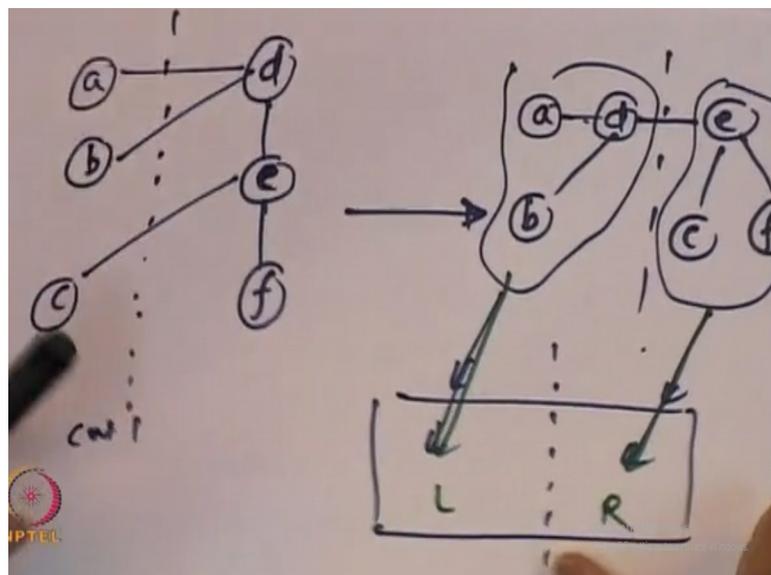
Now over here, I have an example again the details are not too important here unlike it is not an example where we are going to work out delays and some numbers. So here is simple net list AND, okay. Again there are whatever 6 gates and we have been given this chip area which seems to have some locations where standard cells can be kept, in terms of that but the intention is to kind of identify 6 positions, 6 different zones of cell, locations where this 6 gates can be placed in a list cluttered way.

Or, in the more efficient kind of to give us more efficient layout or as efficient as possible. So let us say this algorithm that is based on this idea of partitioning which will repeatedly find minimum cut and used it for kind for the recursive approach to divide the circuit and correspondingly divide the chip in two parts and do this layout generation placement in a divide conquer fashion.

We will be given initial cut there say this is the initial cut, cut one. That it separates ABC from DEF, okay. So this is the circuit with some initial vertical cut and correspondingly let say we have say arbitrary chosen that this part is to be put on the left side and this part is to be put on the right side, okay so ABC should be somewhere and DEF should be somewhere. Now what is next?

So, separately we will be looking at partitioning some one particular example of partitioning algorithm but let say we have some partitioning algorithm available that works with an abstract representation of this circuit, okay.

**(Refer Slide Time: 17:44)**



Abstract representation of this circuit could be like a, b, d, e, yeah, I mean this could be one of the partitioning algorithm like Kernighan-Lin algorithm which does not really bother about the electrical significance of wires, source drivers, directions and so on. Just uses basic essence of the topology. Here it is like a is connected to d, b is connected to d as over here. Who is driving, who is driven is not being represented here.

Just some undirected edges, a connected to d, b connected to d. d is connected to e. d is connected to e. c and e are connected. c is driving e in fact and e and f are connected is driving both the pins on F. But essentially we just look at this, we capture the essential information that is e and c are connected, E and F are connected. Yeah it does not seem to be very satisfactory but still I mean this could be one of the starting points for your investigation.

So and cut one that has been given to us initial cut is reflected as something that separates. Now first we would try to get the best 2 way cut, I mean this graph is very big the best by a partitioning 2 cut. Cutting it into 2 parts with equal sizes and with minimum number of edges crossing is a hard problem. Okay so but in this small example we will be able to work out the best one.

So for example some algorithm will help you figure out that. ADB can be on at one side unlike in the initial partition and CEF on the other side, okay. So initial partition was this, okay but this turned out to be a poor partition. In fact, yes there are 3 edges cutting across. But if you were to use this partition that separates A, B, D from C, E, F and that would be the best one. It would cut only one. That is over here, okay.

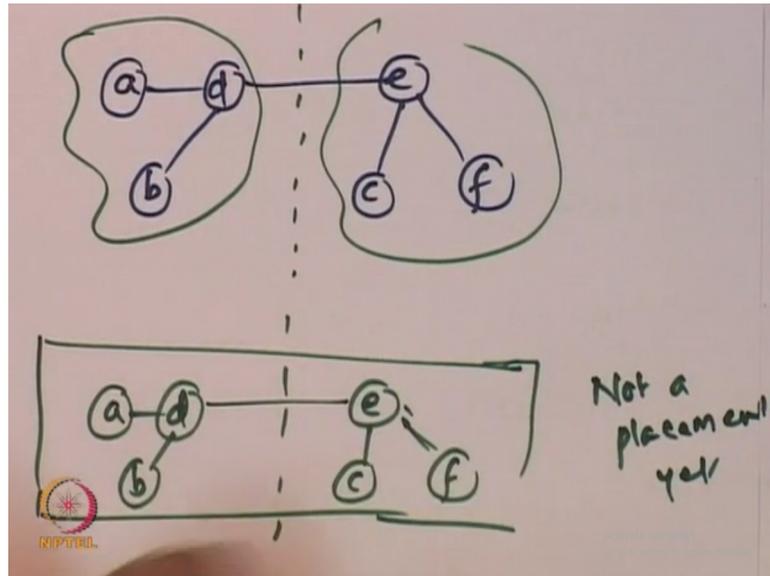
So clearly this is the candidate to be laid out on this partition should be used to like put this portion over here and layout this portion of the circuit over here, okay. We have correspondingly divided as chip into 2 parts let say by a vertical cut line in the left part I will attempt to make a lay out of this and the right part I will try to make a layout of this. That seems them very natural way to go further right.

And again the same these 2 sub problems can again be done in the same recursive min cut based up way fashion. Okay you finding the minimum cut again dividing this into again using vertical cut but may be more natural would be to like do alternating kind of horizontal center

vertical cuts. So in the next stage while laying out this on to the left part or laying out this on to right part we could use horizontal splitting of this chip area by this horizontal cut line.

And obviously there will be more interesting variations of this like adaptations of this. I am not trying to give you the best algorithm but something that explains the idea very simply. Now we have a problem, I mean we have 2 sub problems.

**(Refer Slide Time: 21:58)**



One of implementing A, D, B on 1 and other one of implementing E, C, F and okay. This sub problem, sub problem and this is the best cut that we have. So here A, B somehow we should cook up a good layout for this. We should cook up a good layout for this, okay. This is not a layout but I am just saying that this is to be placed here this to be placed here. This is just not a full layout algorithm just a placement algorithm.

Now and of course this wire have to be placed, this is not a placement, okay. But we have identified this is to be targeted for this, this is to be targeted for this and this wire we will have to be correspondingly laid out during the stitching process during the conquer process. So recursively we go down further and look at the sub problem on. A, D, B again the best cut will be something like this.

This is the best cut and for E, C, F the best cut could be this, okay. There could be other very good cuts but let see this and based on this we will like put mark A and D to be in the upper half. Previously we are divided this chip into by a vertical cut line now we are going to divide

using horizontal cut line. AD and B were to be on this and using this because we are separated ADB circuit into this 2 part using min cut.

I will correspondingly have the split objective, split line objective of making a placement of AD in this part and B on this side. But we should have the wiring to complete the picture and similarly I say C and E are to be put here and F is to be put here and E and F, okay? Now that has been probably decided, we could have been apparently justify to put CE here in the bottom portion and F in the other portion or this way also.

But let say if we had to make some arbitrary decision in the absence of any other hint any further information. That is what I am going to lead on to the so called the idea so called a terminal propagation which is due to Dunlop and Kernighan. So further like, we have now 4 sub problems the sub problem of AD, placement of AD the sub problem of placement of B which is already solved.

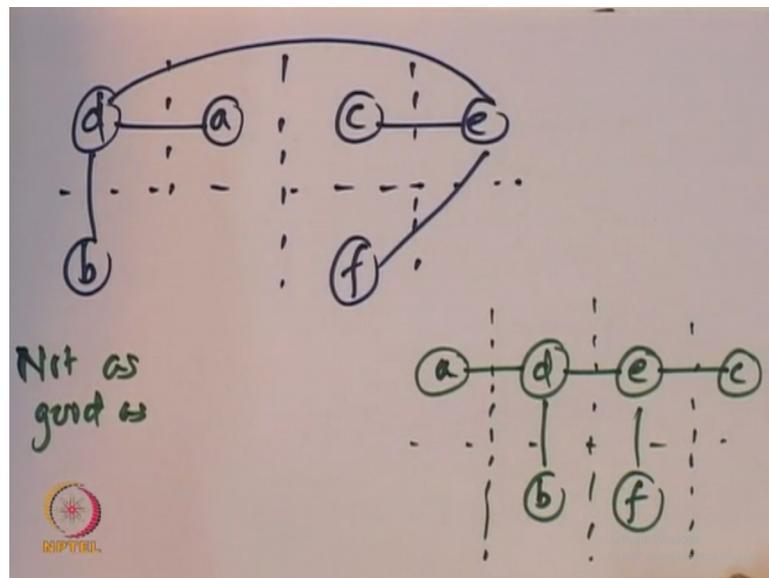
And similarly the other sub problems is of layout for placement for CE. For that again we divide this and this. So this is going to lead you to this color coding is nothing specific but now for in the absence of any other clue we could have place D here and A here and similarly C here and E here, okay and we might have place B over here and F over here, okay?

Yeah, means like, the sub problem about doing the placement of AD in this portion could have further divided until we do not need and to go any further and then we could have may be decided that D is over here and A is over here. So that solves the problem for replacement of AD in this portion of the chip. The placement of these arbitrary decided to be here, placement of F is no contention but say with arbitrary we decided to put it here.

C and E in this same way. Then completing the wiring if when the router completes or we try to estimate the wires symbolical the wires had to be connected this way, C to E and E to F and D to E, right? Yeah but does not look like it is too bad but clearly there is a better looking solution, I mean just the example is small enough just to help you see that we are not arrived at, I mean by doing something arbitrary although we have made use of minimum cut, the best cuts at different times.

There was a plenty of choice in relative locations and that might have you would like to get, like to do something extra to get a better solution.

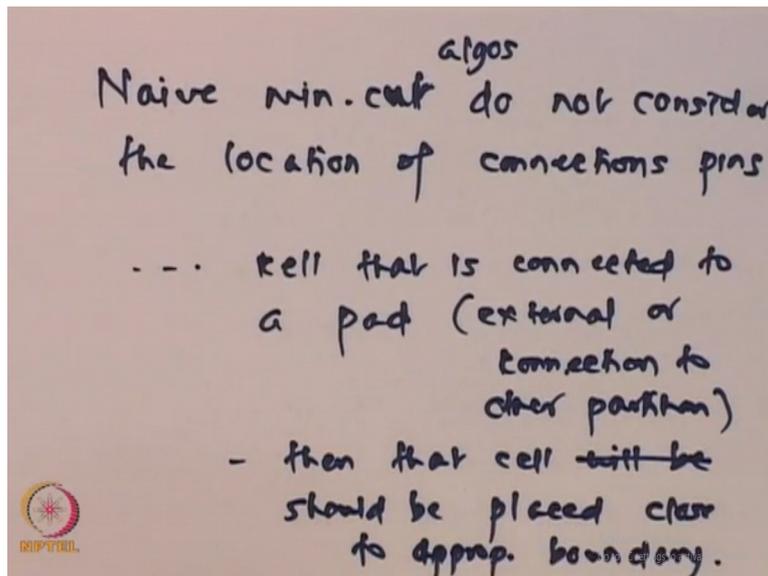
**(Refer Slide Time: 27:49)**



So we have this layout. But this is not as good as, has something that we hope that we could have found A, D, E, C on the same chip. We would have like to arrive at this even for this simple example, I mean and so really what has lead us to this has been bit of arbitrary in the choice of like AD is to be laid out here. I mean it does not matter whether you as far as sub problem is concerned it does not matter whether you put D here, A here or the vice-versa, okay?

So we had taken some arbitrary decision and that lead us to a poorer quality. So we would like this, we would like to be able to make use of some extra information which will help us to some lay out of this kind. That is something that we will motivate the extension of min cut algorithm due to Dunlop and Kernighan.

**(Refer Slide Time: 29:44)**



So remark, the Naive min cut algorithm which simply relies on finding the best cuts and not really like think about other information. Such Naive algorithm does they algs do not consider the location of pins. S, exactly what that means we try to illustrate to example, okay. So, for example nowhere in this algorithm that stimulation that illustration that we did we nowhere can worried about things like if a cell that is connected to a pad external or like connection to other partition.

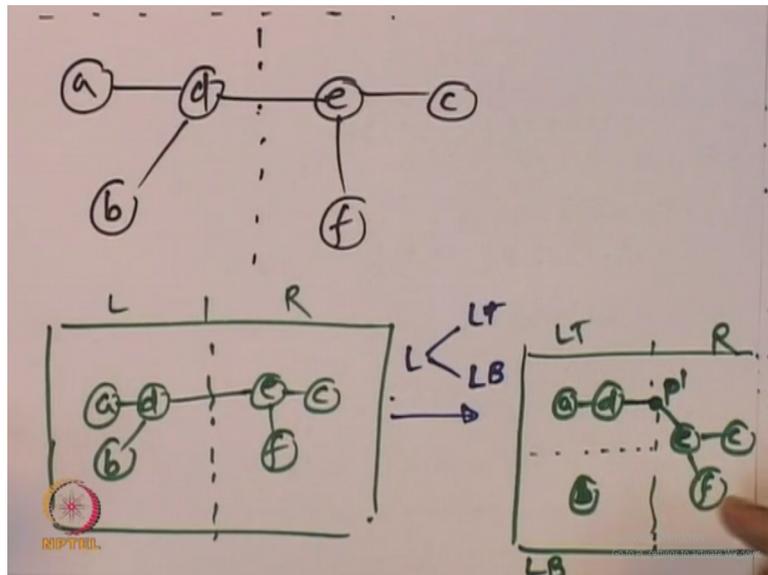
If we have some information about that with this some cell is to be connected to the, this has to be near a boundary, a particular boundary then that kind of cell will be placed or should be placed close to appropriate boundary, right. So, the location of the pad is going to be used in these decisions of placement. So, this kind of like idea did not apparently was not apparently getting used and Dunlop and Kernighan developed this terminal propagation idea to give some do some justice to this, okay as and extend to enhance Breuer's.

**(Refer Slide Time: 31:59)**

Dunlop & Kernighan  
 developed  
 TERMINAL PROPAGATION  
 idea  
 to enhance  
 Breuer's min cut based  
 placement

So, let us take the same example and to highlight the main ideas here again you will have to take things with a bit of pinch of salt. Example is contrived to illustrate some points. We should not try and punch unnecessary holes in that like so it is let me go back to the same example.

(Refer Slide Time: 33:15)



So, we are already had found a good cut, okay for the overall circuit and that is ADB and D and E will be connected. So, in terms of placement problem we kind of committed that this A, D, B we will be put somewhere in this left part and the right part E, C, F. I am deliberately drawing then in a very congested way just to indicate that I have not yet made I have got the input for this sub problem and good input for this placement sub problem.

So, I have the topological information which, so I am just going to as in the initially I had not placement information. So I am going to as if like everything is at the center of the like near the center of this part and everything is going to be, all this are going to be a near center of this right part of the chip that is why I drawn them in a congested way nearby and there has to be this edge. Okay.

Now to do the, while solving this sub problem of getting the placement for ADB we let us say that is, this is important. So, now we are trying to further solve this left sub problem by splitting into left top and left bottom. So, here E, C, F, okay this is E and this is, so here we recall that ADB was split into for like we had for solving the left sub problem we had to make a partition, we had a good partition of ADB

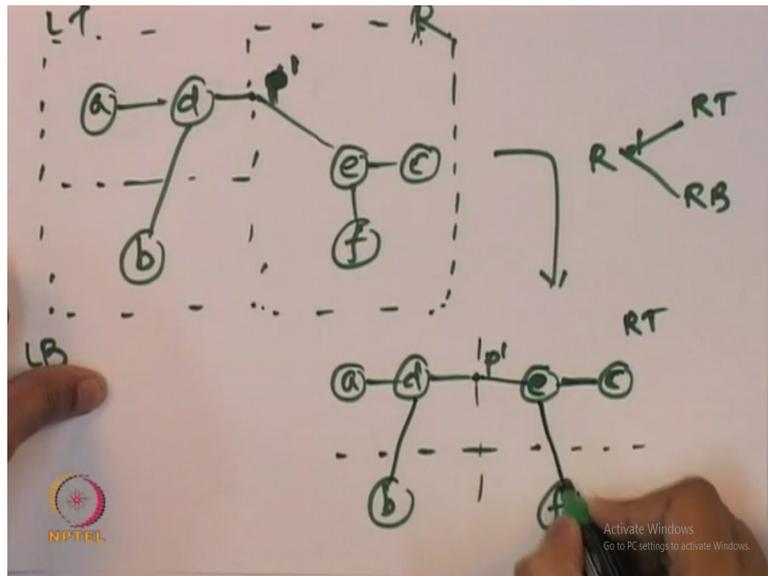
and that turned out to be AD on one side and B on the other side, okay and we decided to kind of put AD on this left top this is LT this is LB and this is R, right part which we are not yet touched, okay. So, now we see that this connection DE is like compared to this where we did not have any like we are not separated ADB in top and a bottom part now that we have separated the connection from D to E will kind of predominantly be here.

So, Dunlop and Kernighan approaches two kind of like whenever you have situation of this kind where things get separated I am not describing it very formally in to like this the fact that AD has been moved here how that fact is to be reflected while solving the right sub problem is done with the help of an artificial connection pin called P dash, okay.

This P dash is included here is kind of introduce here to capture the fact that this net DE is connecting D on this side and E on this side but the D is already placed in that top left, left top part, so P dash is kind of placed at the on this boundary of right side shared boundary but in this upper half, okay. So, with the fixed location of P dash we are hopefully while solving the right sub problem we will be able to put decide on the relative placement of E,

and C more appropriately, okay that is the idea. So, P dash is going to be a fixed location over here for the right sub problems. So this is the way we are kind of propagating some information about, this is not exactly terminal propagation but in general it is to be used for that. We are propagating some information through terminals like artificial terminals or artificial pins, okay.

(Refer Slide Time: 37:41)



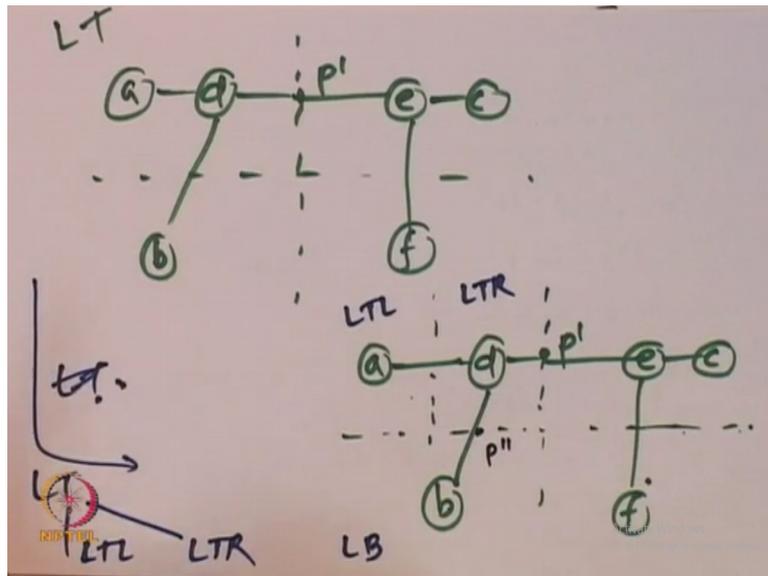
Yes, whether I missed this D to B, okay. So we have, we decided that A and D are going to be in the left top, this is LT and B is somewhere here and this is P dash and this is E, C, F. This is the right sub problem that we have not yet touched but now we are going to address this. So, now while solving the right sub problem we are going to take pay some attention or give some consideration this P dash which is fixed over here, okay.

That is going to be an input, part of the input two the right sub problem. But the main thing is about partitioning this E, C, F. So, E, C, F we are going to split it into E and C and F on the other side but it is going to be clear now that, okay this part is same A, D we had B. So E and C are going to be brought in the upper part because if we were to put E and C, over here it would be reverse right.

Because it will be reverse because E has to be connected to P dash which is in the top part, so E and C both should be here. In fact, we are not making any relative placement of E and C, we are just deciding whether E and C should be in the top part or in the bottom part and whether correspondingly F should be here or here. But it is natural that since P dash is connected to E and P dash is fixed to be in the top part, E also along with C should be in the top part.

So, this is my right top and this is my right bottom. So, I have like kind of done one level of this problem, so I am not fully recursively completed that. So, R has been split into RT and RB, okay?

(Refer Slide Time: 40:11)



Next let us say we focus this is the situation we have A, D. Sorry here, I missing I should not miss this. So, let us say we want to like we solve this problem for LT. So LT is to be split into LTL and LTR. So, I am going to split this LT into two parts, left and right part and correspondingly put A and B place A and B in appropriate portions. So, everything is as it is P dash E, C still unplaced over here,

only thing that we had decided is that E and C are going to be here, F is going to be here, B is going to be here but now we are deciding on the placement of A and D. We are solving this problem and for that we made a partition A and D are to be in separate parts. But I am not clear because of the connection of D with P dash, D has to be over here, okay and A has to be over here. Okay. So, you can regard this as being influenced by P dash in some sense.

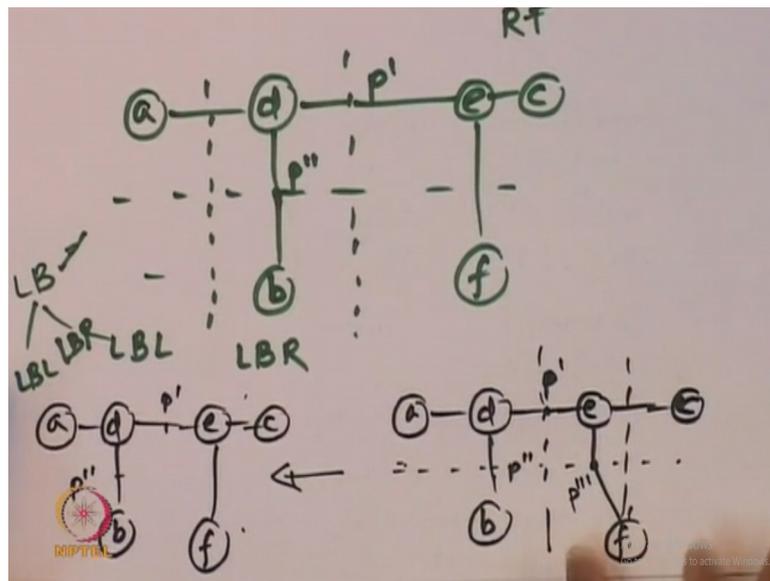
It might be if you do a recursion in different way things may or may not, but this is the main idea when can like in fact you can have your own variation of Dunlop Kernighan algorithm where you make use of this terminal ID, propagation ID in a different way, okay. So, do not assume that I am sticking to exact description of that algorithm. In fact, the idea is to kind of learn this.

So that you can invent you own or discover you own modifications or figure out like alternate approaches and so on. Okay, now this left top problem is completely solved, right. So LTL, LTR left top left, left top right is solved. This is left bottom. Now we will go on to sorting this

sub problem and then later on we will pay attention to this. Okay, now to figure out the placement of B whether in this or this again it will help to think of this as P double dash okay.

Now because D is in the right part of left top it is clear that because of that P double dash is going to be like somewhere fixed along this boundary and because of that B has to be when B has the choice of being here or here left bottom right or left bottom left it would be placed, the algorithm would place it more or like justifiably over here.

**(Refer Slide Time: 43:23)**



So, this is my P double dash and this will force B to be here. We were to decide where B should be and with the help of this P double dash we have decided that B should be over here. This is LBL, LBR, so we have basically solving the LB problem when like making the fixing the layout for this left bottom by like recursively dividing things into LBL and LBR.

A and there is nothing to be done here and B when it has a choice it would based on the location of P double dash it is connected to B will be more logically over here, okay. So, this has solved the left side problem completely. Now, similarly the right side problem, okay. Now let us focus on this R top problem and this will take us to this is fully done with the help of P double dash this is with the help of this.

Now EC has to be, look at this RT, EC has to be split in two parts E and C where should whether E should be here or E should be here C should be here or C should be here. That is again going to influence back P dash. So, E is more logically here and C is over here, okay.

Moment E and C gets separated this way the connection to F gets like biased towards in the upper portion, okay.

Earlier I drawn this connection between E and F to be like this but now I will show it slightly more balanced fashion and I will call this point on the cutline connection as an artificial cutline point called P triple line cut, okay. So, now what is the purpose of this just a way P double dash induce this B to come on this side rather not on this side P triple cut is going to induce F to be placed here not here, okay.

So, from here clearly what we have going to get is A, D, B this must be the help of P double dash this was with the help of P dash E, C, F, okay this is the best lay out that we are hoping to get. So, there is some role of clearly there is some evidence of role of terminal propagation could be exactly like say same as what Dunlop Kernighan suggested or could be your own variation of that.

So, I believe like the main point has been brought out that is with the help of the extra information about terminals initially there could be the fixed location of this particular input has to come from outside world it could which is some other subsystem which is laid out on the left side. So, that input is coming through one of the pads on the left side of the chip.

Similarly, something else could be on the right side or top side based on that the placement of at some stage in this top down min cut based placement the locations, certain decisions will be helped by like this information and we will be able to cut down on the ambiguity and because of which one might get misleading the later stages of the algorithm and this will help as this example obviously has been contrived to show the effect of this modification.

And we have arrived at this good answer compared to bad answer that we did not like over here, okay, idea we would have, we just using the min cut we could arrive at this but instead we have managed to arrive at this kind of solution. Just confirm that it is. So, this is what we have arrived at exactly what we have wanted, okay with the help of P dash, P double dash and P triple dash.

You can, this example is illustrated bit differently as well as there is another example that you would find in the book just to confirm your ideas your understanding and learn more about it

from the book I have avoided to discuss it in a very formal rigid way just kind of focus on the highlights the essence, notions. Thank you.