VLSI Physical Design with Timing Analysis

Dr. Bishnu Prasad Das

Department of Electronics and Communication Engineering

Indian Institute of Technology, Roorkee

Week - 04

Lecture 20

Introduction to Partitioning – II

Welcome to the course on VLSI Physical Design with Timing Analysis. In this lecture, we will discuss about Partitioning. So, the content of this lecture includes hierarchical partitioning. In particular, we will discuss about the chip level partitioning. In the previous lecture, we discussed about the system level partitioning and board level partitioning. In this lecture, we will concentrate on chip level partitioning first. Then we will discuss how we can develop different types of objective function and constraint for different types of partitioning. Then in case of chip level partitioning, we have different types of design styles. So, depending upon the design styles, we have different types of partitioning. We will discuss that in detail. So, whatever we discussed in the last lecture, the hierarchical partitioning, the partitioning is divided into three different categories. One is called the system level, board level and the chip level. So, we discussed this system level and board level in detail. In this lecture, we will discuss about the chip level partitioning. Basically how we should design a chip and we partition the bigger design into smaller sub-circuit and how we can do that.

So, the partitioning of a chip into sub-circuits is called the chip level partitioning. I have a bigger design is there. I want to break that into sub-circuits and we can design each of the sub-circuits independently. The primary objective of the chip level partitioning is basically how I can get a very optimized design of the chip. Very optimized in the sense that at in different angles like speed is the most significant objective, the speed should be improved, then power should be minimum and area should be optimized. So, this speed, power and area all this angle we need to optimize our design inside the chip using chip level partitioning. So, we have different types of design style. So, we have full custom design style, we have standard cell based design style. Depending upon the different types of design style, we have different types of partitioning. So, basically the different types of partitioning algorithms will be applicable for different types of design style. So, here what we are doing is that in the chip performance basically is determined by your critical component should be placed close to each other. So, whenever I am doing the partitioning of the chip, I should worry about the chip speed actually. The speed is basically determined by your logic gates and interconnects. The speed is determined by two components, one is a delay of the logic gate, logic gates in the critical path and the second one is the delay of the interconnect or the delay of the interconnect. So, what we are doing here is that we have already logic gate from the logic synthesis flow and this interconnect connecting them should be as minimum as possible to improve the chips performance or the speed.

So, here what we are doing here is that the critical components, the critical path or the longest path in the design should be assigned to the same partition. Instead of taking that critical components to different partition, you should put all the components critical component in the same partition such that the connection between the partition should not be there. If the critical component is going between the partition, then the speed will be decreased. So, if I have let us say this is the partition 1, I have a partition 2, if my critical component is going between them, then speed will be less. Here the speed will be less.

So here the speed is less. Let us say this is the case 1. So if my critical component is in this line, so what it is telling that if you have a critical component, you allow that critical component in a single partition let us say the either the P1, the critical component path should go inside the P1 or P2. So this is the case 2. So here the speed is improved. So what is the bottom line is that whenever you are going for a partition, do not use your critical component moving from one partition to other as much as possible to improve the speed of your design. Then the system performance basically improvement relies on minimizing the length of the critical path whatever I was talking about. So we need to look for the critical path of the design. Then we will discuss about how we can formulate this partition problem using some kind of graph theoretic notations actually. Earlier we discussed about graphs. So here we are using that graph to abstract out our actual circuits into basically the actual logic gates and interconnects will be represented by a graph. So here you have a, if you have a given circuit is given to you. So we can represent that circuit into a graph G with V, E. The V is basically set of vertices actually which is representing your components. So the vertices are basically your logic gates in case of cheap level partitioning.

So the hyper edges are basically the E1, E2, E3 are the connection between the components actually. So what is a partitioning problem? So I have basically I need to formulate the partitioning problem. So let us say I have a partition V is given and which we divide into V1, V2 and VK. So basically what we are doing is that there should not be any kind of overlap between V1 and V2 or any of them. So we can write that one is basically VI intersection VJ.

There should not be any overlap, should be null. If I is not equal to J and union of, this is V, union of I equal to 1 to K VI is equals to your V. So basically you have a V is there which is containing all the components. So we are dividing into V1, V2, VK means K partitions. Let us say I have a bigger design.

I want to divide it into different, let us say here I divide into three parts. So what it says that VI intersection VJ means there should not be any overlap between P1, P2 or any of the partition. There should not be any overlap. They are disjoint sets. So the first point is that they are disjoints and the second point says that summation of P1, P2, P3 or V1, V2, V3 together creates the overall system, together create the overall PCB or together create the overall chip.

So this is the definition of basically how we can formulate the partition problem. Whenever we have any kind of problem formulation, we want to optimize the problem formulation. So whenever we have optimization problem, there are two things comes into picture. One is called the objective function and the second one is called the constraints, sets of constraints because constraint is not one line, maybe multiple line, sets of constraints. So, objective function can be minimization or a maximization problem.

So the objective function can be a minimization or a maximization problem depending upon the requirement. So optimization is a generalized word. Objective function can be minimized or maximized depending upon the requirement. So what is, let us say I am teaching a class, so my objective is that I want to train all the students and how I can constrain them, taking exams and giving assignments, doing homework, all these are constraints to optimize them, to improve their performance. So whenever we have any kind of mathematical formulation, what we are doing? We have an objective.

If we have constraint, then the tool can optimize that objective function better. Tool will optimize, your mathematical algorithms can optimize with that constraint to get our things proper shape. So basically here the constraints and objective functions for the partitions. So there are two things, the objective function and set of constraint can be different depending upon different levels of partitioning and different levels of design styles. So the level of partitioning means here you have a system level, you have a board level, you have a chip level. So based on different level of partitioning your constraint will be different, depending upon your design style your objective function and constraint can also be different. So here we will discuss the objective first, so we are discussing the objective function first, then we will discuss the some of the constraints. So the objective function here is to interconnection between the partitions. So whenever I have two partitions, the interconnection between the partition is called the cut and now what is our aim is to minimize that cut. So main objective here is to minimize the number of cut.

So this is my objective. So how I can mathematically express that using this formula. How whenever basically I have two partition is there, so one partition one line is going to the another partition another line. So this first summation is talking about one partition, second summation is talking about the another partition and if I have a connection from one partition to the another partition, then I have a cut is introduced in this objective function. So if I have a line going from one partition to the other partition, then Cij is 1 otherwise it is 0. So take an example here, so our aim is to minimize the number of cuts. So take example here, so here if you can see I have a cut line and the number of basically the cuts are here is 4 in this partition, this is one example of a partition. Let us say this is the partition P1 which divided into block A and block B, it has 4 cuts. It has cut equals to cut is 4. Then if you can go here, I have two cuts. Let us say this P2, this partition has two cuts.

So which is the best partition because since I have less number of cuts, my objective is to reduce the number of cuts, so the P2 is a best partition because number of lines going from block A to block B is 2 here. Then we have another objective, so this is a objective function, this is a second objective. So here what is the objective is to basically the delay between the partition is significantly larger than the delay within the partition at any level of partition. So if I look into the delay, inside the partition the delay is small, outside the partition if I am going the delay is large. So my objective is to take less number of signals between the partitions.

So what is happening is that here you have basically that signals going from one partition to the other partition, basically those critical signals delays should be minimized. Let us say if it is going through multiple partitions that should be minimized. Let us say I have one partition P1, then another partition P2, then another partition P3 is there, then a signal is critical signal is going through this. So I need to take the maximum path delay, this max is corresponds to maximum delay from here to here inside the module, inside the partition and outside the partition then I need to reduce that, minimize that. So the max of Pi belongs to P, h of Pi should be minimized. So where P equals to basically P1, P2, P3 is a set of hyper path and Pi is the number of times a hyper path is cut. So now number of terminals is a constraint actually, it is not a objective function is a constraint at any level, any level means either the system level or a board level or a chip level, the number of terminals for a partition is a constraint for the partitioning algorithm. So you have a upper bound on the Ti, so Ti is the highest number of terminal possible for each partition, I need to choose something lesser than that or equal to that one. So this is basically a constraint, number of terminal count for a given partition Vi. So now we have basically area of each partition, basically I have three different types of partition, three different types of basically system level, board level and chip level.

So in case of a system level area of each partition of the board is fixed, so the area appears as a constraint actually. In case of board level it is also important to reduce the area of each partition, basically each chip to reduce the cost of the board actually. So in case of chip level the size of the partition is not so important as long as the basically the partition is balanced actually. So we need to choose the partition such a way that basically each of the partitions should have equal area, balanced area. So now area of a partition basically it has something called upper limit and a lower limit, within that upper and lower bound we need to implement the partition. So we have a upper limit on the area of partition, you have a lower limit of the area of the partition. So basically max and min area of the partition is used to optimize the area of the partition. So this is go inside the optimization algorithm in order to optimize the partition. So also there is another concept called number of partitions. So let us say I have a bigger system how I can determine how many PCBs are needed or how many chips should be used in a PCB.

So all these depends upon the number of partition. So number of partition is a constraint at the system and the board level. So basically whenever I have a bigger system I should not use many PCBs and each PCB having too many chips actually. So that will increase the overall cost of your system. So in case of chip level number of partition is basically determined by the placement algorithm which is used because what is the problem with the chip level is that chip area is fixed and it has some of the hard IPs area is also fixed. So based on that we need to determine the partitioning area. What is the available space is there based on that we should fix the area of each partition. So number of partition is determined by the placement algorithms actually how it can be placed it has the capability to place the design in the chip level. So it is a constraint actually. So how many partitions should be used? The constraint is the number of partition is basically it has a minimum number of partition and it has a maximum number of partitions.

So the Kmin and Kmax represent the minimum and maximum number of partition allowed for a given circuit which basically determine the total how I can optimize the overall design. Basically two things whatever we discussed the objective functions and constraint will be different based upon the design style. The design styles includes your full custom, semi custom in case of semi custom you have a standard cell based design we have a gate array based design we will discuss that which objective function and constraints are used there. So in case of full custom design style we have basically partition can be of any size so because we are designing the thing manually so here the layouts is done manually. Layout is done manually. So what is the advantage is that? So layout is done manually we can determine the pin count how I can determine the pin count? The pin count will determine let us say I have a one partition is there. Let us consider one side of the full custom design. So let us say this distance is let us say 10 micron just giving an example. Let us say my metal one I am creating the pins in the metal one.

I have another metal one. So this is two metal one it is two pins. So the metal one to metal one actually the distance the pitch I need to find it out. Middle of the metal one this is metal one this is another metal one. So middle of this one to middle one middle of this one I need to take it out. So this middle of this one let us say it is two micron. So the M1 pitch is two micron. M1 pitch I need to write that pitch. So now how many pins I can put it? So here in two basically pitch is two micron I can have two pins. So for 10 micron I can put basically 5 pins. So number of pins number of number of terminal or pin whatever the same thing terminals are equal to basically from here to here I am talking about how many terminal can be done there. So that is 10 micrometer divided by 2 micrometer. So it will be 5. Similarly we can also find the number of terminals or pin of the entire design by taking the perimeter of the design into account. So the minimum spacing between the two adjacent terminal is called this terminal pitch or the whatever I talking about the metal one pitch. So that will determine my number of pins in case of a full custom design step. So since there is no area constraint we can determine the perimeter of the design based on our requirement.

So here in case of partitioning algorithm using full custom design style I have to satisfy the objective one. The objective one and the constraint one and constraint two. What is the objective one? Here the objective one basically this one objective one. Objective one is basically the interconnection between the partition. So I have less number of cut. So this is I need to satisfy and constraint one number of terminals and the area constraint. These two things I need to consider in case of full custom design step. Objective one that is the basically the number of cuts should be minimum. The objective one is basically is related to number of cuts should be minimum and the constraint one is related to the number of terminals and then the constraint two is basically the area it should be within certain area bound actually. So in case of full custom design style we need to take these things into account. However if I have a high purpose full custom design then the speed should also be important then the objective two is related to your delay. So that need to be considered. So in case of high performance design we need to consider the objective two your delay should be minimized and in case if there is it is not a high performance design then the cut between the two partition need to be minimized. So now if we look into the standard cell design style. So here your overall circuit is used using some partitioning algorithm what we will discuss in subsequent classes actually. So basically your overall design is partition into sub partition and each of the design is implemented using the cells in the library and if the libraries are simple contains very simple cells then partitioning algorithm has to optimize the objective one basically partitioning algorithm optimize the objective one and the constraint one and constraint two. The objective one is basically reduce the number of cuts between the partition between the partition the constraint one it should be within the terminal counts it is related to number of terminals the constraint one is related to number of terminals and constraint two is related to you have a area constraint area of each partition. So area of the each partition should be bounded by area min and area max of that partition. So the area min and area max will be given by the designer when we are going to create the partition. So if your library have many different types of cells some of which may be complex then the partitioning problem will be little bit complicated because different cell has different area. So if you have complex gates are there your partitioning problem will be more complicated. Then we have gate array based design style where we have basically we partition the circuit using bipartiate partitioning basically whole circuit is divided into two part recursively until basically each of the partition will fit to one of the gate array. So here we are following a methodology of bipartitioning approach basically the it will minimize the number of nets between the partition boundary. It will basically optimize or minimize the number of interconnects between the partition boundary.

In this lecture we discuss about chip level partitioning we discuss about objective function and constraint for the partitioning problem at different level of hierarchy and we also discuss about your design style will also determine how I can set my objective function and constraint. For example we have full custom design style we have semi custom or standard cell based design style we have gate array based design style each of them will have different objective function and constraint based on the requirement. So those will be considered while going for different types of design style.

Thank you for your attention.