VLSI Physical Design with Timing Analysis

Dr. Bishnu Prasad Das

Department of Electronics and Communication Engineering

Indian Institute of Technology, Roorkee

Week - 04

Lecture 19

Introduction to Partitioning – I

Welcome to the course on VLSI Physical Design with Timing Analysis. In this lecture, we will discuss about partitioning. Partitioning is one of the first step of any kind of system level design or any kind of board level design or any kind of chip level design. So, the content of this lecture includes the introduction about the partitioning, then some of the terminology involved in partitioning, then we will discuss about what is the hierarchical partitioning. Whenever we go for any kind of partitioning, we can formulate that problem into using some mathematical functions. So, we will discuss about different problem formulations, how we can do to use that problem formulation for optimizing the partitioning that we will also discuss in detail.

Then different types of design styles depending upon the different types of design style partitioning can be of different types. So, first of all we will discuss about what is partitioning, basically what makes partitioning important in this VLSI physical design and why we should go for partitioning to improve the productivity of our design things. So, the partitioning is basically is a process of breaking down a complex system into smaller units which can be manageable, we have a bigger system, we can break down into smaller part which can be implemented easily. So, the process of decomposition is basically done using hierarchical approach such that each subsystem which can be implemented in a smaller area.

So, here let us say if you go for a chip level partitioning, let us say if I have a bigger chip and I want to design a processor, basically what I need to do inside a processor I have a control unit, ALU, pipelining stages. So, I need to divide each one of them in such a way that I can design independently and combine them in a top level which can be done parallely. So, here what I am telling is that decomposition is basically dividing the whole system into smaller subunits. So, whenever I am doing the partitioning basically, so it is one of the critical step in the design process which allows efficient implementation of complex system. So, if you have a very complex system, how I can break that into sub

blocks and each of the sub blocks can be divided independently and that can be combined in the top level.

In the same point, I can highlight our previous discussion on algorithms. So, the algorithms, there are several types of algorithms is there, there is an approach called divide and conquer. This partitioning is quite similar to the divide and conquer approach of the algorithmic implementation. So in case of a bigger algorithm, what we do? We divide into sub problems and combine them later. Same thing what we do in the hardware side, we also break a bigger design into subunits and implement them independently and combine them.

So, let us take some example to understand the partitioning in detail. So, here if you can see the top level design, you can see multiple logic gates are there. So, the number of input is basically 48 gates. I want to break that into multiple subunits such that I can independently implement all the three modules separately and combine them. So, if I divide, this is the original design, then I want to break that into three partitions, three basically units. So, here this is a partition 1, this is partition 2, this is partition 3. So, if I divide them into three different partitions, so on an average, each one should occupy at least minimum 16 to make each of them equal size. So, but sometimes it is not possible to make equal size, so we can allow some little variation around 16. So we have basically on an average around 16 we need to divide it. So if you can see here, you have a block A, the block A has 15 logic gates and the block B is having 16 logic gate and the block C has 17 logic gate.

So, if you can see that on an average, the number of logic gate in each partition is almost close to whatever we need 16. So it is quite equally partitioned. And then there is a one of the most important point is that whenever you do the partition, the number of interconnect coming from one partition to other partition is important. If you can see here the partition P1 to P2, we have four lines are there. This is the one line, this is the second line, this is the third line and this is the fourth line. So, the cuts are four. Similarly, if you can see the other from partition P2 to P3, you have four lines are there. This is first line, second line, this is the third line and this is the fourth line. So the cut one has four lines, cut two is also having four lines. So this is an example of a partitioning. So here what can be done is that somebody can implement P1, somebody can implement P2, somebody can implement P3 parallely and then you can combine them together to make the implementation process faster. This is one of the goal. And it has multiple goals, we will discuss one at a time. So this parallelism of doing the work parallel is one of the goals. Then let us say I take a very small circuit which is having total eight number of logic gates, 1, 2, 3, 4 like that eight logic gates.

I can create multiple different partitions then which is a good partition. How can I distinguish which is a good partition? So let us say this circuit is there and I did a cut here

which cut is let us say this is one type of partition. In the previous example I have only one type of partitioning is given but here what we are doing is that I have a circuit, I can show you two different types of partitions and then I can distinguish which is the best partition among them. So this is a cut CA and if I can do this kind of partition, so this is a block A which belongs to one partition then the block B belongs to another partition or block or partition is same thing. So here you have a block A or partition A, block B and partition B and each one of them having equal number of gates that is 4 and if you can see here number of cuts between the partition here it is 1, 2, 3 and 4 number of cuts here is 4 and if I do a cut like this then what is the number of interconnection between two blocks? I have a block A or partition A, I have block B, the partition B then I have each side this is one goal then whenever you do the partition number of gates in each partition should be same or close to half.

So here if you can see the 4 gates is there in block A, 4 gates are there in block B. Now if I look into the number of external connection between the block A and block B here it is 1 and 2. So then we can decide which one is a good partition. Obviously this second partition, this partition is a good partition because I have only two cuts are there between block A and block B. So number of lines or interconnect between the two blocks if it is less that is best.

So here it is 2, hence the second partition is a better partition. So we will discuss about what is the benefit or the advantage of doing the partitioning. So what it does is that basically whatever I discussed earlier whenever you have a bigger design if one person can do all the work it will take let us say 1 year. Then if I do breaking of that bigger design into smaller unit and I can employ more people to work in the subunits then I can do the parallel work. So it allows parallel work on the subsystem. So if I do the parallel work that one year work can be done in a lesser time if I have more people working in the subsystems. The subsystems at each of the subsystems is done by different peoples. Let us say I have a bigger design D I break down into D1, D2, D3. Earlier let us say the bigger design takes done by one person it takes one year. If I divide into D1, D2, D3 I can employ three different persons.

So on an average the total time will be reduced by 3 at least it may not be exactly 3 because it will take some time to integrate the three modules. So it takes little higher than the 4 months period because I need to also integrate D1, D2, D3 together to create the overall system. So it allows parallel work on the subsystem that is one of the benefit. Second is the it reduces the number of interconnection between the subsystems. So whenever we are doing the partitioning we do not do partitioning randomly we have to do with some kind of intuition that number of interconnection between the partitions should be as minimum as possible without compromising the speed of the design.

So we need to reduce the number of interconnection between the subsystem which will basically minimize the complexity of the overall system. So this is the second advantage of partitioning and the third advantage is basically if you have smaller design it is basically it is easy to optimize compared to optimizing a bigger design. So it is simple and efficient design process because I have a small design I can optimize it better and it saves time and resources. So it is very beneficial whenever you have a proper partitioning of the bigger system. Now we will introduce some of the terminology which is used in case of partitioning. One is the cell actually. What is this cell? Cell is basically a logic block, logic gates, the logic gates or the functional unit like ALU or memory unit or some other blocks. So those are called the cell actually. So cell is any logical block or logic gates or functional unit built from a component. Actually you have a bigger unit so these are the individual units that is used to create the components. Then partitioning, what is a partition or a block? Both are the same thing. Basically if you have a collection of components or the cell is called a partition. For example, these are 1, 2, 3, 4, 5, 6, 7, 8 are the cell. These logic gates are cell here, are cells and your block A is a block A or B are the partitions. Block A and B are the partitions. So the partition is a grouped or grouped collection of components and the cells.

So it is a collection of components and the logic cells, logic gates. So now we have a K-way partitioning problem. Basically let us say I have a basically very big module. For example, if you can go to this slide here, I have a very big design. I divide into 3 partitions. So it is a 3-Way partition. I have P1, P2, P3 are the 3-Way partition or if I have K blocks that is a K-Way partitions. So basically the partitioning problem divides the overall block into K partitions where K can be anything more than 2. K can be anything greater than equal to 2. So if K equal to 2, then it is called bipartiate partition. So if K is greater than 2, then that is called K-Way partition. So K should be greater than equals to 2 and if K equal to 2, that is called bipartiate partition and if K is greater than 2, then it is called K-Way partition. Now we will introduce the term called cut and cut set. So the cut is basically an edge between two nodes I and J. So it is an edge or a line or a wire between two nodes I and J if I and J belongs to two different partition A and B. Let us say I have one partition A. I have another partition B, block A or block B, partition B, the line between them are called a cut. The lines between them are called the cuts. These are called the cuts. And the collection or the set of all cuts is called the cut set. Let us say this is one line, two lines.

So the collection of all the cuts are called the cut set. In each iteration of the algorithm, I need to check how I can reduce a cut and the number of element in the cut set should reduce. Let us take an example here. I have a circuit here. So this circuit is easy to represent using graphs what we discussed in our previous discussion. This circuit is represented using a graph. So if you can see here, I have the graph G1 is basically containing the nodes 2, 3, 5. 2 is belonging to this NAND gate, 3 is belonging to this

NAND gate, the 5 is belonging to this NAND gate. So this G1 is basically contains the nodes 2, 3, 5 and the graph G2 contains 1, 4, 6. Then I need to check what are the cuts there and what is the elements in the cut set.

So the cuts here are basically this line. This is one cut, this is another cut. So the first cut what is there, so that is between the node 1 and 2. So this is one of the cut actually. This is cut 1 and this is the second cut 5, C is a cut 2. So let us say you denote this one by cut 1 and you denote this one by cut 2 and the cut set will contain the cut 1, cut 2. So this is all about the terminology. So then we will discuss about hierarchical partitioning. The hierarchical partitioning is basically we let us say we can divide this partitioning problem into different levels. We can have system level partitioning, we can have system level partitioning, we can have board level partitioning, we can have chip level partitioning. In case of system level partitioning, whenever I am building a bigger system and we need to think how can I divide the system into group of PCBs. PCB stands for Printed Circuit Board. So the PCB is the Printed Circuit Board. Let us say if I can open the computer basically that unit, I can see multiple PCBs are there which has to build your complete system of your computer or a server or something. So you have one board is called your motherboard. There will be something called your main memory. So all are independent PCBs. They come together to build your complete computer system. So that is called system level design. So your memory is separated, your motherboard is separated, your multiple data boards come together to build your computer system. So that is an example of a system level partitioning. So we have multiple PCBs are needed to build a system.

So how I can break the things to different PCBs that problem is called your system level partitioning. Now we can talk about board level partitioning. Let us say if I want to design a PCB, how can I place my chips such that my size of the PCB will be smaller. So here what is happening, I am designing a PCB. So the PCB has multiple chips or ICs or integrated circuits, passive component, all these things together I need to build the board or PCB. So how I can place, how I can decide the chips that is your board level partitioning problem. Now comes the chip level partitioning problem which is we are most worried in this course basically that is called the chip level partitioning. Whenever I am going to design a chip, how I can break the complete thing to smaller sub-circuits. So I have a bigger design, I want to break that into smaller sub-circuit and such that each one of them can be optimized separately and put it together to build the chip which is optimal in all directions. So here this is an example. So you have a system, this is your system. Now inside that system you have multiple, if I open that box, let us say this is a computer or some system, I have multiple boards are there, these are my PCBs, these are my PCB and this is one PCB, this is one PCB. If you can see inside that PCB you have multiple ICs are there. Then this is one example of IC, so these are your IO which is the only thing which can talk with the external world. So these are the input output pins. So this input output pins is the only thing we can have connection from inside of the chip to the

outside world. So that is the interface between your internal of the chip to the outside world. So whenever we design the chip we can decide how many IO's can be there. So it has some upper level constraint and it has a lower level constraint. So I will discuss that whenever I am going to the chip level partitioning. So now we will discuss each one of them separately. The first we will start with the, we will discuss in the top down format. So we will start with the system level partitioning then we will go till the chip level partitioning. So in case of partitioning basically a system, we are dividing the system into a group of PCBs. So the complete system is implemented using a group of PCB and each PCB has some constraints actually what are the constraints involved in a PCB? It has some dimension, the size actually, size of the PCB and it has a number of pins, number of terminals for the connections.

So it has limited number of terminals for connections. So whenever you have a system level design and number of terminals are determined by the system bus connected between your different PCBs actually. You have these terminals are having system bus, basically the system bus size will be determined by the number of terminals actually which will connect from one PCB to other PCB. Then each board has a specific terminal count. So based on our requirement what are the important pins which need to be connected to the other critical pins, important means critical pins that need to be connected based on that we will decide the terminal count. So which will basically limits its connection with the other board.

Basically the number of terminal count is also important whenever each of the PCB is designed. So that needs to be determined when you are designing each of the PCBs. Then one more thing is that reliability of the system inversely proportional to the number of boards of the system. If I have number of boards is higher, reliability reduces but it depends upon how the systems are integrated or communicated with each other. So if you have a number of PCBs are there you need to put more effort in communication between the PCBs such that the reliability of the overall system can be improved.

So here basically our prime goal is whenever you are doing the partitioning how to reduce the number of boards of the PCBs to implement the overall system. So whenever I am going to design the complete system my first goal is to how to reduce the number of boards which indirectly improve the reliability of the system. So then we need to do some kind our another objective is to optimize the system performance. So overall system performance, the speed and power dissipation of the overall system all these should be optimized whenever I am going for the system level partitioning. Then comes the basically system in case of system level partitioning we have the speed is most important thing.

Whenever the signal is traveling between the PCB. So I have two types of delay basically delay inside the PCB. So delay whatever the performance or the delay. I can delay or the

performance will determined by the signal traveling between the PCB and the signal traveling inside the PCB. So what inside a same PCB. So basically if I can see basically off board delay, off board delay is this one the first one delay outside the board is called the off board delay plays the important role in determining the system speed actually because the off board delay is larger than the on board delay.

So this one is the on board delay. Then I have third category of delay is the on chip delay inside the board you have chips the delay is very less inside the chip. So this is the on chip delay. So this delay is small actually this is small and this delay is large. So since the delay at each step is important in order to determine the system performance we which is the largest one I need to optimize it optimize that to improve the overall speed of the system. So we should take significant amount of care to optimize the delay of the signal traveling between the PCBs.

So now I will go to the board level partitioning. So the board level partitioning is basically partitioning of a PCB into chips actually this is called a board level. I have to design a let us say microcontroller board. I want some minimum number of pins whatever needed for my system. I need to have a basically microcontroller I have some memory of chip memory flash memory is needed.

So the memory I need to add now I need to add some power management IC. So all these things should be determined in and how I should integrate them in the board level to optimize my overall system performance. So I want to design a PCB for a microcontroller I need to choose which microcontroller is suitable for my design. I need to choose which kind of memory is suitable for my design. I need to choose which power management IC is sufficient to provide power to my overall system and also the third one is your communication. Communication means you need to have different types of communication among the board like USB you have this is Ethernet all these need to be determined before I want to design the PCB for a microcontroller.

So basically the point here is that I need to decide the chips how I can partition the PCB into chips so that my board level partitioning is successful. So main aim is to minimize the area of each chip as the total manufacturing cost of the overall board is proportional to the chip area. Even if I have different types of microcontrollers are there they can have different types of packages. So if I use a BGA package versus QFN package the BGA package has more number of pins. So those even if whenever I want to determine a microcontroller I need to think which kind of package I should choose for my microcontroller so the overall system performance can be optimized.

Number of pins in the board will be optimized. So whenever I am doing board level partitioning basically the signal traveling between the chips and the ICs actually and through the connector which often has a high resistance the connectors are high

resistance that contribute significantly to the signal delay. The signal delay means that the critical path delay in the board actually. The signal traveling between the chips and through the connectors will if it is coming in the critical path of my design then the overall speed will be sacrificed. So we need to look for the critical path in the design such that those paths should be optimized properly. In the board level how can you do we cannot do inside the chip we cannot do because we buy the chips from the market but we can have lot of flexibility how we can route the interconnects inside the PCB.

So, the interconnect routing will also impact our speed of our design. So the critical signal should be routed with proper care to reduce the overall signal delay inside the PCB. So hence reducing the number of chips used on the board will improve the reliability and system performance. So whenever I choose the number of chips let us say if I have two functionalities together I should use two functionality together chip bed compared to using two chips in the board. So there are two things involved here. Let us say whenever let us say I have a microcontroller I will take an example here I have one microcontroller is there another microcontroller is there microcontroller with Wi-Fi chip Wi-Fi module. So let us say I design one board PCB this is PCB 1 where I have to use one microcontroller IC and another IC for Wi-Fi module. So then I create the overall board. This will obviously occupy more area but let us say nowadays some microcontrollers are there which has microcontroller and Wi-Fi module together in the single chip. So we can use that PCB 2 design where I can use a single IC where I have microcontroller and Wi-Fi module is together in a single chip.

So that one will be more optimized save the board area compared to the PCB 1. So obviously cost of the IC will be more but think is that the size of the board will be reduced. So in case of PCB 2 two things will happen one is reduce number of terminals advantages are for PCB 2. Then second one is size of the board will be smaller size of the PCB will be smaller. Since I am using one chip for the overall implementation the speed of the design speed of the design yield is improved. And demerit these are the advantages of the PCB 2 and what is the basically limitations the cost will be little higher.

Similarly I can give an example of a microcontroller with ADC and without ADC. So let us say I have a microcontroller. I want to design a system which has microcontroller and ADC separately. I have to buy the ADC IC I have to buy the microcontroller IC. So the PCB 1 correct. So the PCB let us say this PCB 3 because I have already used PCB 2 this is PCB 3 and the PCB 4 I have a microcontroller with plus ADC.

So it depends upon your application some cases the PCB 3 is better some cases PCB 4 is better. I can give you an example is that whenever I have a high resolution ADC is needed the PCB 3 is better because normally the microcontroller with ADC always use a low resolution ADC like a 12 bit ADC. But if I my application requires basically high resolution ADC then the PCB 3 in that one I have to buy a extra ADC which is high

resolution let us say 24 bit or 18 bit ADC to implement the overall system compared to use the microcontroller plus ADC together to optimize the basically the size of the board. So some cases the two different ICs are also better because of the system requirement.

In this lecture we discuss about what is partitioning and we discuss about basically the benefits of partitioning and we discuss about what are the different types of partitioning like system level partitioning, board level partitioning and chip level partitioning there are three different types of partitioning is there. Then we discussed in detail about the system level partitioning and its advantages then we discussed in detail how the PCB level or the board level partitioning can be done.

Thank you for your attention.