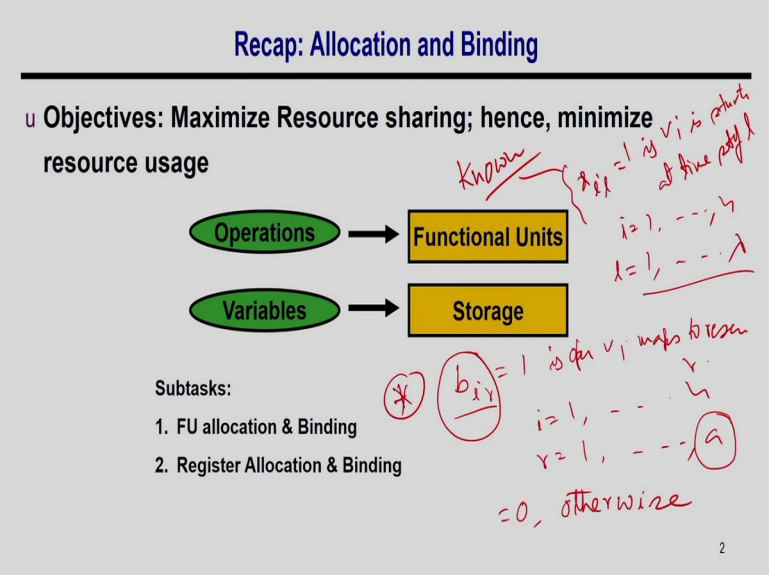
C-Based VLSI Design
Dr. Chandan Karfa
Department of Computer Science and Engineering
Indian Institute of Technology, Guwahati

Module - 05
C-Based VLSI Design: Allocation, Binding, Data-path and Controller Generation
Lecture - 18
Resource Allocation and Binding: ILP based formulation

Welcome back to the class, in this particular class we are going to discuss the ILP formulation for Resource Allocation and Binding problem ok.

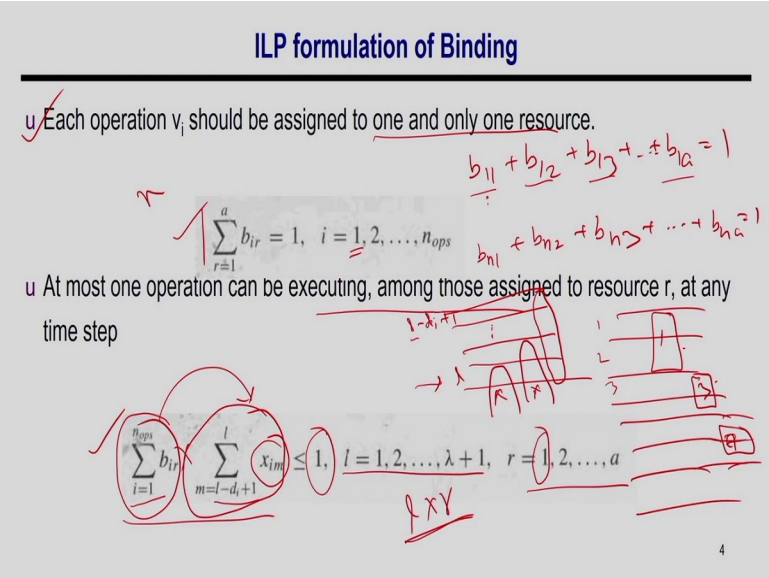
(Refer Slide Time: 00:55)



So, we have already discussed that this resource allocation and the binding problem is mapping the operation into function units and variables to the storage and the objective is to minimize the number of function units needed and the storage needed right. So, first you have to allocate the number of FUs and storage and then you have to bind each operation to the FUs, each variable to the storage ok.

And we are particularly discussing the FU allocation and binding in this class ok. So, we have seen that this problem, in general, is np-complete and then we found a left edge algorithm that can actually solve this problem efficiently for a particular class of problem when the conflict graph is an interval graph ok.

(Refer Slide Time: 01:37)



In today's class, I am going to see how we can formulate the ILP - Integer Linear Programming for this binding problem ok. So, for this what we do is basically decide 2 variables. The first variable is b_{ir} right. So, I have given. So, what is the problem? I have given a set of operations of one type that we have already discussed that we will be going to do the individual operation at a time say I am going to execute only some multiplier and then ALU and so on ok.

So, the $b_{ir} = 1$ if operation v_i maps to resource r right. So, where my i = 1 ,….., n and r = 1,…….,a right so; that means, I have 'a' number of resources I am assuming and 'n' number of operations and I can map ideally any operation to any resource. So, that is why I take a variable Boolean variable called

$$b_{ir} = 1, \; v_i \text{ maps to r}$$

$$= 0, \text{ otherwise.}$$

So, in the ILP formulation what I am usually we do? We find out all the constraints needed and we should have an objective function and what this ILP does? It tries to map the values to this unknown variable such that all the constraint gets satisfied and whatever the objective function that is achieved ok.

So, in this case, what is our unknown variable? This b_{ir} is the unknown variable because we do not know which operation is going to map to which FUs ok. And in some cases,

this ‘a’ may be also an unknown variable because if it is not a resource constraint scheduling, then this ‘a’ is not known to me. our objective is to find out the minimum value of a. In some cases when say we solve the problem of MLRC that Minimize Latency under Resource Constraint; that means, that time I have given the upper limit of resource right in that time this also is a constraint because I know what is the number of FU is needed there ok.

But what is the problem then ILP solving that time? It actually gives this b_r value I know the upper limit of the FU available, but you tell me which operation is going to map to which resources ok. So, this is a variable that is unknown to me and I am actually going to find out the value of this using ILP ok.

So, there is one more variable that is needed is $x_{i\ l}$ which we have already discussed during the ILP formulation of the scheduling problem. So, $x_{i\ l}=1$, v_i starts at timestamp l right. So, here i can vary from one to n, I am assuming there are n number of operations and a l is varying from one to lambda. So, lambda is something the number of timestamp is available to me the latency of the design ok.

So, this is something that determines the start time of an operation if it is a single cycle that is the end time also, but if it is a multi-cycle it will start from the timestamp l, but it will go for l + 1, l + 2 to some l + di- 1. So, if the delay of that operation is d_i ok. So, that is already known to me.

And during this allocation and binding problem this is a constraint it is known right this is known to me because the scheduling is already over I know which operation is scheduled where, but I am going to use this variable for the formulation ok. So, what is the constraint let us try to find out the constraint? The first constraint says that you have the operations right I want to map them to FU. How I am going to map this? I am going to make sure that each operation is going to be executed and can be mapped to only one and only resource ok.

So; that means, one operation I am going to map to only one of the FU not 2 multiple FUs which is called logical right. So, this all operation is going to execute only once right, and just mapping them to one FU is sufficient right and that is also needed. So, this is the first constraint. So, what is saying that here that summation b_{ir} for each i each type of operation for all resource available is 1.

$$\sum_{r=1}^{a} .b_{ir} = 1, i = 1, 2, \ldots, n_{ops}$$

So, suppose if you are operation is 1; that means, b_{11} + b_{12} + b_{13} …….. +b_{1a} =1. That means operation 1 can be either map 2 resource 1, resource 2 or resource 3 and resource 4 similarly I have to do for operation all operations. So, then let us say this is the n^{th} operation, then b_{n1} +b_{n2} + b_{n3}+……..+ b_{na} =1; that means, operation n is either going to be mapped to resource 1 or 2 or 3 or . . . . . or a; that means, at most one of this variable will be 1 right because I make the summation 1.

So, it will ensure that each operation is going to map to only one FU ok. The next one is. So, it is a hardware resource you assume that there is a FU there is a multiplier in one clock it can execute only one operation although I map 3 multipliers to that FU that particular FU or multiplier I can only execute one of the assigned operations in a particular clock right.

What does it mean? Although I can assign three multipliers to this operation say for the first two cycles I am going to do this operation 1. So, suppose this is multiplied 2 cycles and say operation 3 and 4 I am going to do operation 3 and say for operation 7 is going to map for timestamps 3 and 4.

But at a time I can only execute one of the operation because it cannot do more than one operation at a time right. So, that also to be assumed right how I can do that? Given by this constraint let us understand. So, the first thing is that what is try to find out? Let us assume that this term gives the operations which is live at timestamp l it is going for all timestamp right.

So, I am going to for timestamp 1, I will make sure that only one operation will map to a FU, I will go to timestamp 2 I am going to make sure that only one operation is executing in timestamp 2, in timestamp 3 I will make sure that only one operation is executing in timestamp 3 and so on right.

So, how can do that? I have to identify all the operations which will be live at a particular timestamp how can I find that? So, if I am in set timestamp l. So, if the operation starts from here then it is also live right. So, suppose this is an operation is multiple cycles

right. If it starts the previous cycle and still going on then also it is live here right or it can go till the step and it is actually finishing here right.

So, then where it can start? So, l – d_i +1, d i is the delay of the things. So, if it is starting here then is going to this will be the last of step because it is a d cycle operation ok. So, all these cases this operation is live l right. So, this summation tells that if I am in timestamp l you check is there any operation started l-d_i+1 till l. So, this is the

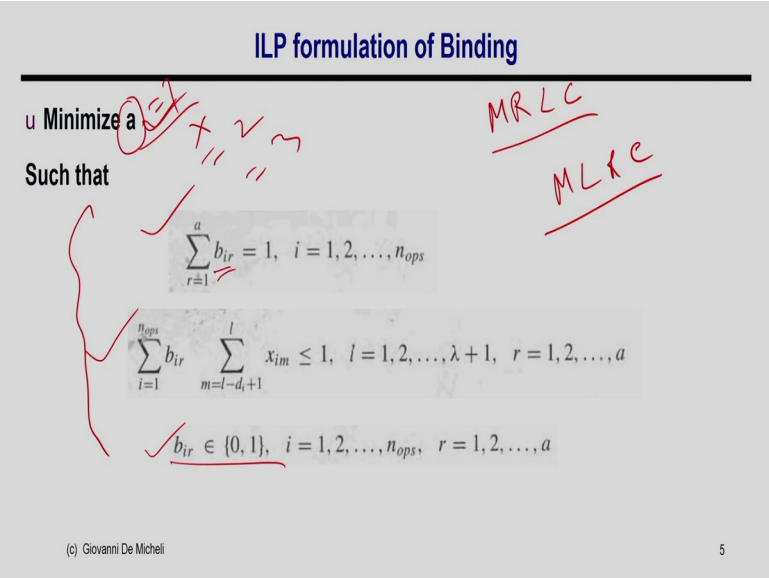
Σ from l - d_i +1 till l is there any operation which started there? Then $x_{i\,m}$=1 in that step right if it is started there and the delay is d_i then the operation is live.

And then I am going to consider. So, this is this test the all the operations that are live in a particular timestamp and then what I am multiplying with that? I am multiplying with that operation that are actually all the operations that are mapped to this particular resource right I am taking a resource at a time.

So, I am going to consider resource 1, I will identify all the operations that are going to map to resource 1, I will identify all those operations that are mapped to of resource 1 and I will check among these operations how many of them actually live in a particular timestamp l and this sum multiplication should be <= 1. So, it makes sure that at most one of the live operations in every timestamp is executing in that in resource r right and that r can vary from 1 to ‘a’ and this I will go for all timestamps.

So, it is basically how many constraints will be there? The constraint will be l * r. Because for each timestamp for each resource I will get one such constraint and how many this kind of resource will get? This I will get for each type of operation. So, it will be n right. So, I will have n number of constraints here, l into r number of constraint here and that is sufficient right for mapping.

(Refer Slide Time: 11:10)



Then what is the ILP formulation? So, if I assume that my ‘a’ is not known it is basically minimization of resource under latency constraint scheduling, then I need to identify a right. So, then my objective function will be to minimize a subject to the unique binding unique execution of the operations and the Boolean is the b_{ir} is equal to Boolean variable right.

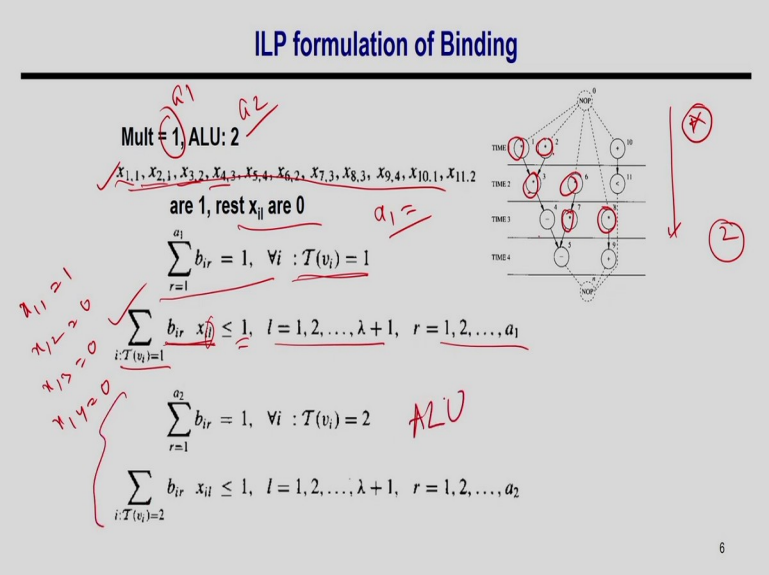
So, if I give these things to an ILP solver, it will give me the value of this b_{ir} such that this is the minimum value. For this MLRC where R is known the resource is known I do not need this right. I just give this constraint and I just ask the ILP solver to give a value that satisfies all the constraints because this is known to me right some constraint is given.

So, then what this ILP solver does, it actually only gives me the mapping of this operation to the resource available ok. There can be third use of this. So, I do not know ‘a’ and I want to see whether ‘a’ is say I will try to find out the minimum value of ‘a’ and how can I find that.

Say I make sure that I will just put is, is it possible to map all the operations to only one FU? And if the solution exists it will give you this is the mapping if not it will say the constraint is not satisfied; that means, given the constraint I cannot execute everything in FU 1 only one multiplier right.

So, then I can increase it to 2, and I can ask the question to the ILP solver again. If it can give a solution; that means, 2 is the minimum number if not then you can go for 3 and so on right. So, either I can give this constraint minimize or I can just check this one by one and check what is the minimum value right. These is all three-way I can actually utilize this ILP formation ok.

(Refer Slide Time: 13:04)



Let us take an example. So, we have this schedule which we used to take even throughout this discussion and for this I am going to take only the multiplier right. So, for many multipliers are there? 1, 2, 3, 6, 7 and 8. You can see here that since a maximum of two operations are executing in one clock I need 2 multipliers right we have to figure it out also 1 and 2 cannot be mapped to the same FU and so on that will we will see.

But we know that start time ok. So, $x_{1,1}$ because one is scheduled here, $x_{2,1}$ because operation 2 is scheduled in timestamp 1, $x_{3,2}$ operation 3 is scheduled in timestamp 2, $x_{4,3}$; that means, operation 4 is a schedule in timestamp 3. So, basically, these are the variables which is 1, and rest of the X_{il} is 0. For example, for say operation 1 what are the variables available? x_{11}, x_{12} because it can schedule in timestamp 1 2, x_{13}, x_{14} right because there are 4 timestamp. So, this is 1, this is 0, this is 0, this is 0.

Similarly, for each operation, there are 4 such Boolean variable is possible only one of them is one and what are the one is mentioned in this list ok. So, these are the 1 and rest

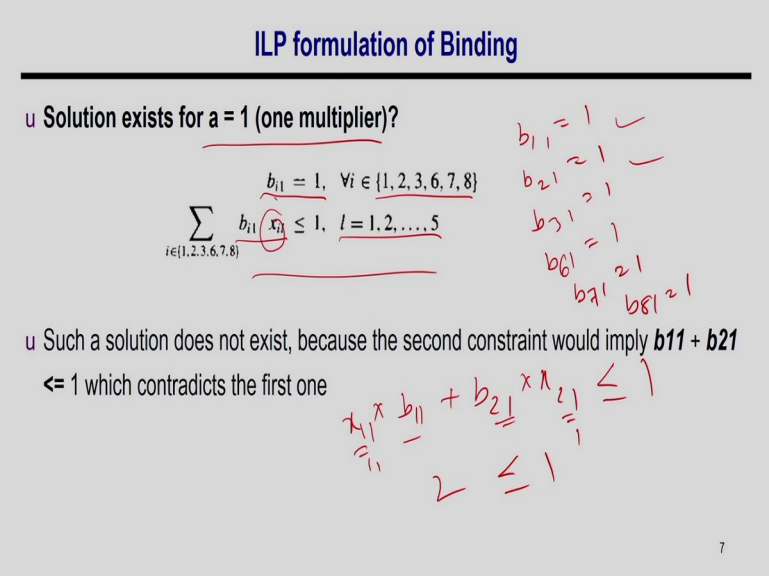
of the 0 because this is a constraint given to me ok and I am assuming that my multiplier is type 1 right its a type 1 ok.

So, now what are the constraints? So, I want to see that for. So, this is a_1, a_1 is the limit of the multiplier right. So, I want to find out the limit a_1 and this is a_2. So, for all operations which is type 1; that means, their multiplier their Σ r=1 to a_1 =1 so; that means, they can be mapped to only one FU right.

So this is already discussed and similarly, this is the constraint for this second problem that for each because here this all operations are single cycle I do not have to find out that x that complex second stamp, I just say that all the operation that are mapped to the multiplier and then every timestamp l.

So, the number operation is executing is less than equal to 1 and this is for each timestamp and each type of each number of resource right. So, type is a multiplier and each instance of the multiplier and this is the constraint for ALU you can get them easily ok.

(Refer Slide Time: 15:32)



So, now let us say I ask the ILP solver is its problem can you solve the problem for one multiplier ok? So, if there is a 1 multiplier what are the problems? What is the constraint we have? So, this b_{il} = 1 what does it mean? That i is the operation number right so; that

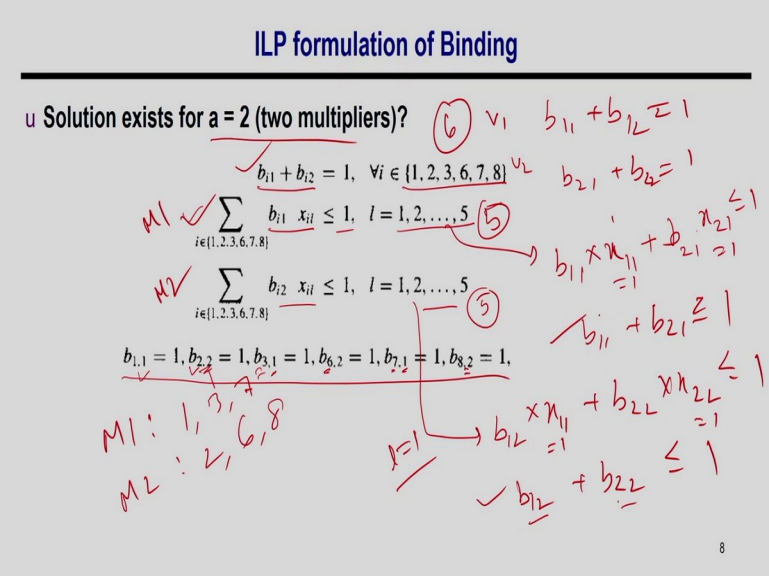
means, operation 1 because only one instance of the multiplier is 1. So, this b_{11}= 1, operation b_{21} =1, operation because these are the multiplier right 1, 2, 3, 6, 7, 8.

So, then b_{31} equal to 1, operation 6 mapping to multiplier 1, b_{61} =1 and b_{71} =1 and b_{81} =1. So, that means, each operation exactly we know while it maps because there is the only one multiplier everything has to map to one multiplier. And now what is the second constraint? It says that every time because the now l is 1 to 5, the operation that are executing in timestamp 1. So, how many operations are executing operation 1? So, operation 1 is say these 1 and 2 right multipliers.

So, the operations executing say b_{11} +b_{21} because operation 2 executing in timestamp 1 and this operation 2 executing time using the resource 1 and this is all this x_{il} = 1 for these two operations right. So, x_{11} *b_{11} + x_{21} * b_{21}<=1 where x_{11}=1 and x_{21}=1 because this I know this is executing in timestamp 1, but I know that b_{11} =1and b_{21}=1 .

So, this is 2 <= 1. So, it will say that your constraint is not satisfied for multiplier one multiplier and there is no solution exist and while which is quite obvious here we know there are two multiplication operations is happening in one clock. So, I cannot do this thing by using one multiplier right. So, this solver will give you no feasible solution exist ok. So, now I will take say two multipliers right.

(Refer Slide Time: 17:43)



So, then what are the constraints? For each operation either it will map two multiplier 1 or multiplier 2. So, that is why I have written this for each i. So, its basically b_{11} + b_{12}=1, this is for operation 1 ,v 1, then for operation 2 it is b_{21} + b_{22} = 1 and so on.

So, basically, it will be either map to multiplier 1 or multiplier 2, this is for 1, 2, 3, 6, 7, 8. So, there will be 6 such constraints ok. And now this is the constraint for this in one time one multiplier can execute only one operation. So, for each type 1. So, say this is for multiplier 1, this is for multiplier 1 because there are two multipliers. For multiplier 1 if these operations are mapped to multiplier 1, then all the operations running in timestamp 1 I mean timestamp l should be less than equal to 1 right.

So, you will get how many constraints here? So, because there are 5 timestamps. So, there are 5 constraints I will get here and for multiplier 2 I will get 5 constraints. So, there are 10 constraints here and 6 constraints here. So, total 16 constraints I will get. And if you try to underwrite this. So, it basically says that all the operations which is say for timestamp 1 I know only 1 and 2 is executing. So, what will happen? So, $b_{11} * x_{11} + b_{21} * x_{21} <= 1$. So, what does it mean?

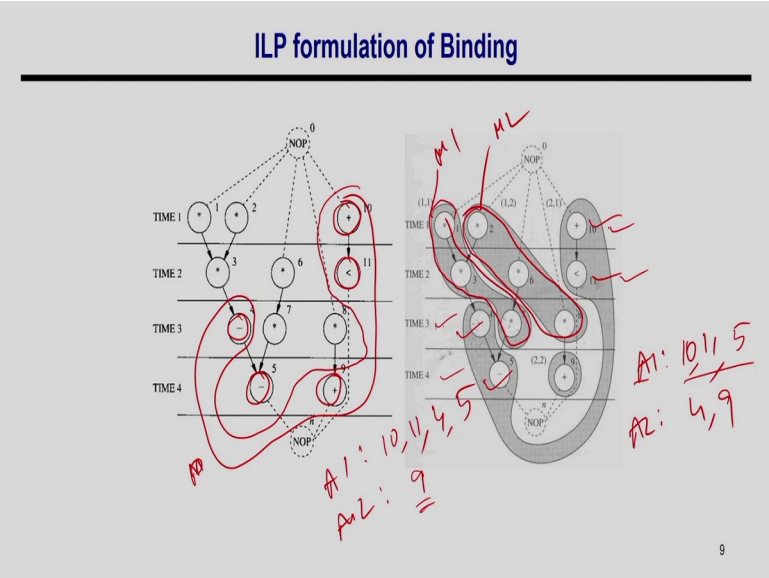
So, either so, this is already 1 I know, this is also 1 I know so; that means, $b_{11} + b_{21} <= 1$; that means, only one of them can be mapped to multiplier 1. Similarly, if I just write for this one the first 2 for timestamp 1 it is basically $b_{12} * x_{11} + b_{22} * x_{22} <= 1$. So, here this I know 1, this is I know 1 so; that means, $b_{12} + b_{22} <= 1$ what does it mean?

So, either operation 1 or operation 2 can be mapped to multiplier 2. And if you just combine this and this it will give some value of this b_{11} and the mapping, I will understand either this operation will be mapped to this or this in multiplier 1 or 2 or an operation 2 also I can map to either multiplier one or multiplier 2.

So, this is how I can write for all timestamps, here I just wrote it for l=1 timestamp 1. So, I can write the same thing for timestamps 2,3 and so on right. If I just write all those 16 constraints and just ask the ILP solver if the solution exists is there any mapping b_{ir} value which will satisfy all these constraints and then it will say this.

It says that operation 1 is mapped to multiplier 1 operation 2 is mapped to multiplier 2 because this two is the multiplier number, operation 3 so; that means, in multiplier 1, 1 is mapped, multiplier 2, 2 is mapped 3 mapped to 1, 6 mapped to 2, 7 mapped to 1, 8 mapped to 2 right. So, these are the value is 1 rest of the b ir equal to 0 right. So, I can understand that I need 2 multiplier which is I mean giving a solution that 1, 3 ,7 is going to execute using multiplier 1 and 2, 6 and 8 is going to execute using multiplier 2.

(Refer Slide Time: 21:26)



So, if you just see here. So, you can see that 1 3, and 7. So, 1 3 and 7, this is the 1 3 and 7 and this is the 2 6 and 8. This is what the multiplier 1 this is my multiplier 1, this is my multiplier 2 and which is correct because you can see here there is no conflict right this ILP also gives me the correct solution.

Now, if I try to solve the problem for ALU how many ALUs are there? So, this is one ALU, this is one ALU operation, there is one ALU operation here and there are two operations since there are two happening in parallel I mean I need at least two ALU right. So, that A1 and A2.

And you can if you can write the same constraint again the way I have developed for the multiplier, you can write that each ALU is an operation mapped to only one of the ALU and for each timestamp, each ALU should execute at most one of the operation right. And if you solve this problem it will give you it will say that you need 2 ALU and then it will give one of the solutions that you map this 10, 11, 9, and 4 into 1 ALU which is possible because these 9, 10, 11 this 5 and 4 because these. So, these are all in different cycle right. So, this is in timestamp 1, this is timestamp 2, this is timestamps 3 and 3 this is 4 right.

So, I can map all these 4 operations 10, 11, 4, and 5, I can execute using ALU 1 and I can use A 2 just to execute operation 9 ok. So, this is the solution that this ILP will give you actually there are many such mapping is possible, but every mapping is as long as

that particular mapping satisfies the constraint the compatibility or the complete constraints, it is a correct solution right.

Once we look into the data path generation probably we will look into this if we just map these 4 operations into FU, there will be a lot of multiplexer sizes will be more because there will be 4 inputs to the multiplier right you have to multiplex the inputs right. So, there is one ALU that is going to execute 4 operations; that means, what? In every cycle, you have to multiplex the inputs.

That in timestamp 1 you give the inputs of, this operation in timestamp 2 you give this operation of, this operation in timestamp 3 you give the operation of this and in timestamp 4 you give the operation inputs of this operation right. So, what does it mean? You have to multiplex the inputs for every timestamp.

So, the multiplexer size will be more and you can see here this A 2 actually remains ideal for most of the time ok. Probably I could have done this that I can do this 10, 11. So, this is also a possible correct mapping that 10, 11 and say 5, I am going to do this and then say A 2 is going to do 4 and 9.

So, then probably the load for this of this multiplier reduce by 1 ok. So, once you construct the data path probably you have to look into this mapping you try to do the mapping in such a way that your multiplexer size is less right. So, at that time probably we have to add some additional constraints to this ILP solver. So, that it actually understands that and makes sure that its operation gets evenly distributed ok. So, those things we will discuss in subsequent classes ok. With this I conclude today’s discussion.

Thank you.