INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

NPTEL

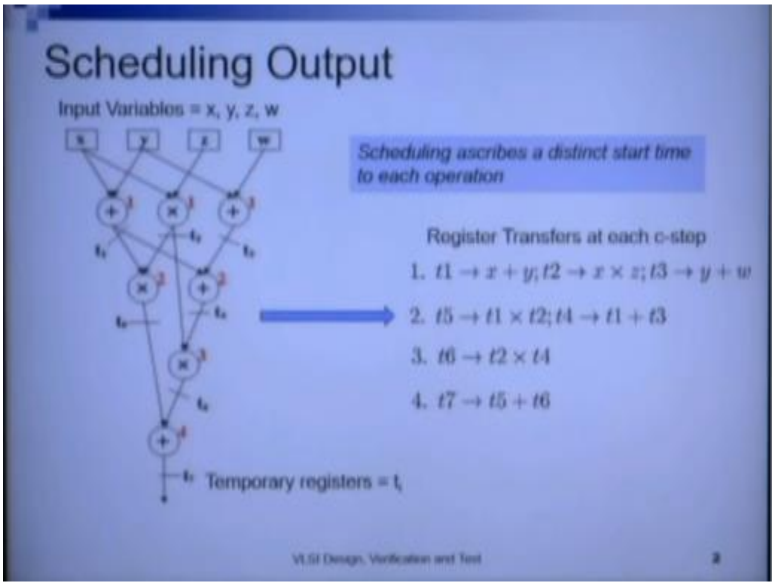
NPTEL ONLINE CERTIFICATION COURSE
An Initiative of MHRD

VLSI Design, Verification & Test

Dr. Arnab Sarkar
Department of CSE
IIT Guwahati

You welcome to module 1 of lecture 6 of the course VLSI design verification and testing the last few lectures we took a look at resource constrained and time constraints scheduling strategies in high-level synthesis in this lecture we will look at an important post editing step allocation and binding.

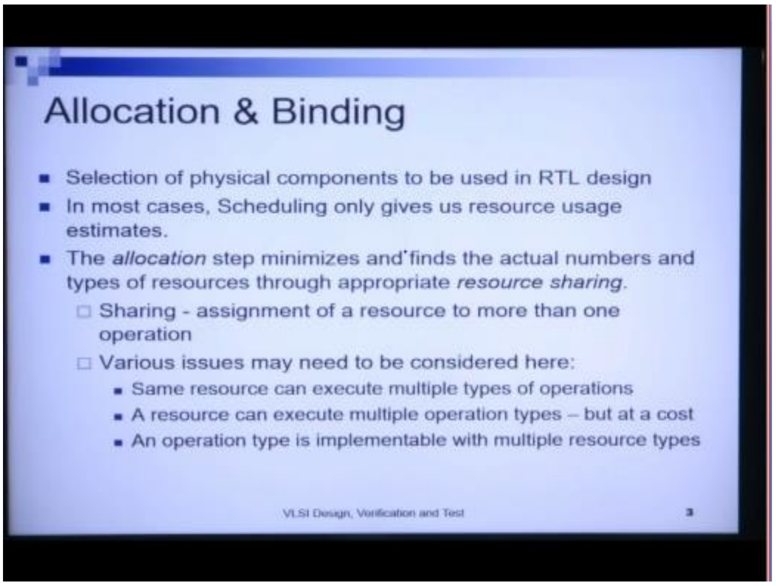
(Refer Slide Time: 00:45)



So what do we have at the output of the scheduling step at the output of the scheduling step we have distinct time steps assigned for each operation. So we have a scheduled operation constraint graph here we have an arbitrary operation constraints graph. In this graph we can see that the operations plus turn multiplication and added here is performed in time step one, in time step two we perform this multiplication and this addition in time step three we perform this multiplication and in time step 4 we perform this addition.

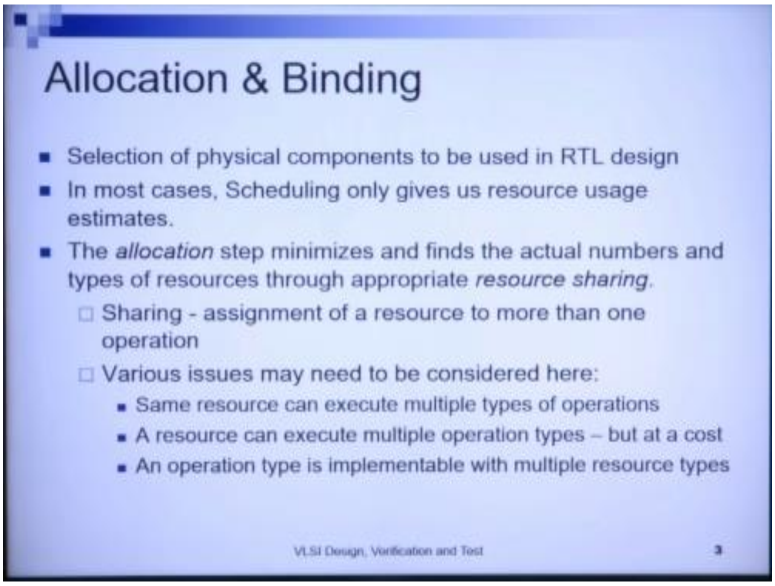
All these are unique time operations each of these operations take input from certain input variables and outputs on temporary registers or temporary variables so the these t1 t2 t3 are temporary variables and what we have is a set of register transfers. The set of register transfers here are t1 equals 2 x plus y in the first time step in t2 equals 2 x into z in the first time step similarly we have different register transfers at each time step at the output of the scheduling step.

(Refer Slide Time: 02:09)



Now after we have assigned a time step to each operation our job is to actually allocate and bind physical components to the RTL transfers to the operations we need to assign functional units to the registers temporary registers we need to assign actual Hardware resource registers we need to steer the outputs of the registers to the appropriate functional units and again we have to drive the output of the functional units to appropriate registers to affect each register transfer. So therefore we need to assign physical components to the register transfer level design

(Refer Slide Time: 02:59)



Now in most cases scheduling gives us an estimate of the resource usage, so in latency constraint scheduling and time constraints and a resource constraint scheduling that we had looked in scheduling every all these scheduling algorithms took an estimate of the resource usage a bound on resource usage bound on latency and provided as a schedule.

Now the work of the allocation step is to minimize and find the actual numbers and types of resources that will be required and this will be done via appropriate register sharing resource sharing, what is sharing? then resource sharing the assignment of a resource to more than one operation now when we do resource sharing we need to keep various issues in mind for example let us consider the case of functional units so the same functional unit can execute multiple types of operations, say that means that in the in the shade during step we saw that the comparison operation the plus operation and the minus operation on all being perform during the ALU.

So I can have multiple operations being scheduled by the same resource and then I can have a resource that can execute multiple operation types so then we are saying that multiple resource the same resource can execute multiple types of operations now here the same resource can execute multiple types of operations but it comes with a cost for example let us say that we have we can combine a multiplication and an addition as an example however the addition the combination of addition and multiplication comes with a cost because it requires a more complicated functional unit safe.

So we can have simple adders that can perform only additions simple multipliers that can perform simple multiple simply multiplications and I can have combined adders and address as well as multiplied which can do both but possibly these combined adders and multipliers take more area than simple adders and simple multipliers now if the objective of your resource binding is to arrive at a minimum area a minimum area design then what do we have here.

Now because we have a combined multiplier and adder then these two operations become compatible meaning that if multiplication and addition is being performed at two different time steps then such multiplication and operations can be combined and implemented using a same resource instance however this comes at a cost that means that what will be the minimized area design will depend on how many combined multipliers adders you are using how many multipliers you are using and how many adults you are using let us say previously you used only multiple only simple multipliers and simple adders.

And therefore previously you could not combine multiplication and addition operations in the same resource or functional unit however now you are being able to combine multiplication and multiplication and addition in the same functional unit. So what you are having now is additional resource sharing now higher resource sharing means I am re using the same function unit more that means I can possibly reduce the area of the circuit however what will be the effective improvement now due to higher resource sharing I am probably.

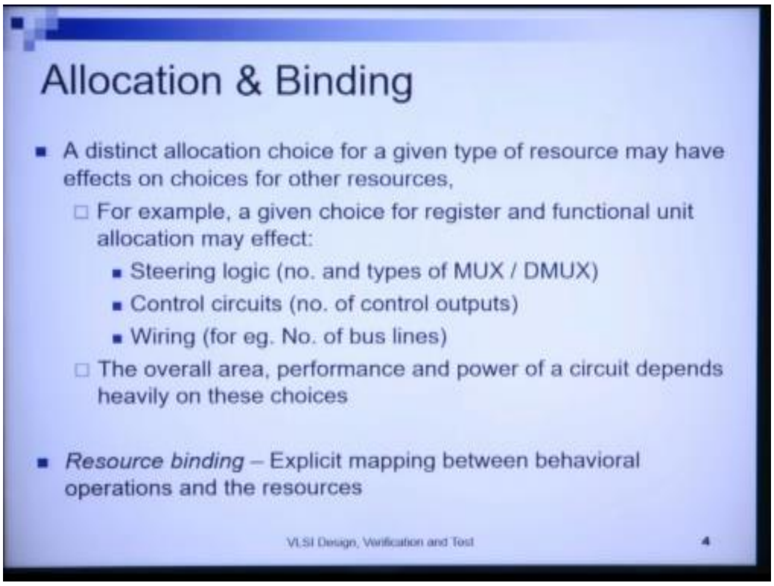
I have probably due to higher resource sharing I have probably reduced the number of simple multipliers and simple adders let us say the total area cost or due to this reduction this reduced the reduction in area due to the reduction in simple multipliers and adders are is X however there is an increase in area because we said that this combined multipliers and adults take a higher area than simple multipliers and adders.

And let us say this increase in area due to this combined functional unit is why then what will be your effective gain in terms of area it will be X minus y so we have to find out so there are issues what I am trying to point is that are issues choice of what allocation step what allocation choice you will have what associating you will have what resources you will choose for which operation is a complicated problem.

This is what I am trying to point to another example is that an operation type is implementable with multiple resource type so the same operation type is implementable with multiple resource types. For example let us say you have two different types of adders for doing the addition operation now added a is hired in performance than adder by but added a also takes hired area than adder by now given a resource assignment problem with a bound on latency as well as your area the choice of the numbers of adders adder a and a derby in your design will depend on how you optimize both area and performance area and latency.

So again what I am trying to point out is that the problem of allocation that is finding the appropriate and correct numbers and types of resources for each operation for the different operations so that my design objectives are met is a complicated problem to solve there are a lot of choices to make.

(Refer Slide Time: 09:44)



Now a distinct allocation choice for a given type of resource may have effects on choices of other resources now I have chosen let us say man for example I have chosen a number a certain number of registers for implementing for allocating the variables in my schedule and the temperature registers in my schedule and I have also decided on the number and types of functional units that I will use to implement the operations in my schedule.

Now given a choice a particular number of registers and particular allocation of the registers and a particular type and number of functional units for a given decision you are steering logic that is the numbers and types of Max's and de Max's will depend for it what will these Max's and deMax's do Max's arbitrator write accesses to form a functional units or and registers on the other hand d max deMax's arbitrate read accesses for from functional units and registers.

So the choice of which register which are variables you have put on which registers which operations you have mapped to which functional units will that will decide on what will be the number of muxes and what will be the types of Max's a 4 is to 1 mux or an 80s to 1 mux the choice of types and number of marx's will depend on the choice of registers and functional units you have made it will also make it will it will also differed determine.

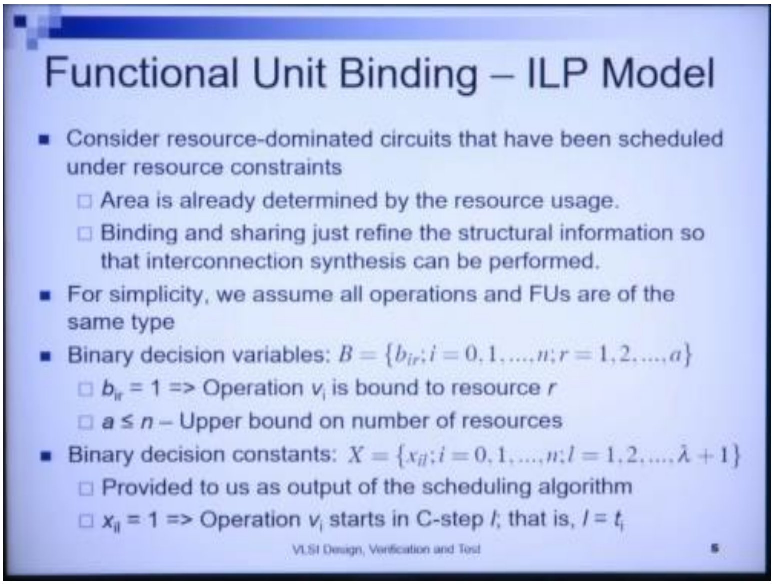
The control circuits that you have for example now the number of control out puts that you will that you will produce will depend on the types of numbers of Max's and de Max's let us say the Select inputs of the muxes and deMax's are controlled by this controllers so when the number of Max's and de Max's change numbers and types of Max's and deMax's change the control circuitry that the number of outputs that the control unit will have to produce a teach time step that will also value.

So the register and functional unit allocation step will also determine what will be the complication or in the control circuit it will also it will also affect the wiring in the circuit what will happen let us say you have buses you have buses in your circuit and your registers float their output on the buses your functional units also float their outputs on the buses now a bus line can only carry a single variable a single data at a given time so depending on the numbers and types of registers Max's functional units what will happen is that your number of bus lines will be determined.

And hence a choice of registers and functional units will also determine what will be your wiring how complicated will be your wiring right so therefore what we conclude from this discussion is that the overall are a performance and power consumed by the circuit will depend heavily on the choices of allocation that you make and hence the allocation problem is an important problem to solve as optimally as possible with this introduction we now define resource binding what is resource binding result binding is the explicit mapping between behavioural operations and resources.

So you can have many instances of these resources for example let us say we have five multipliers in a design multiplier 1multiplied to multiply 3 multiplied four and multiplied five each multiplied say multiplier one is an instance of a resource multiplied to is an instance of a resource and each resource instance multiplier one let us say will be mapped to a number of behavioural operations at different times now resource binding is therefore the explicit mapping between these behavioural operations and resource instances

(Refer Slide Time: 14:30)



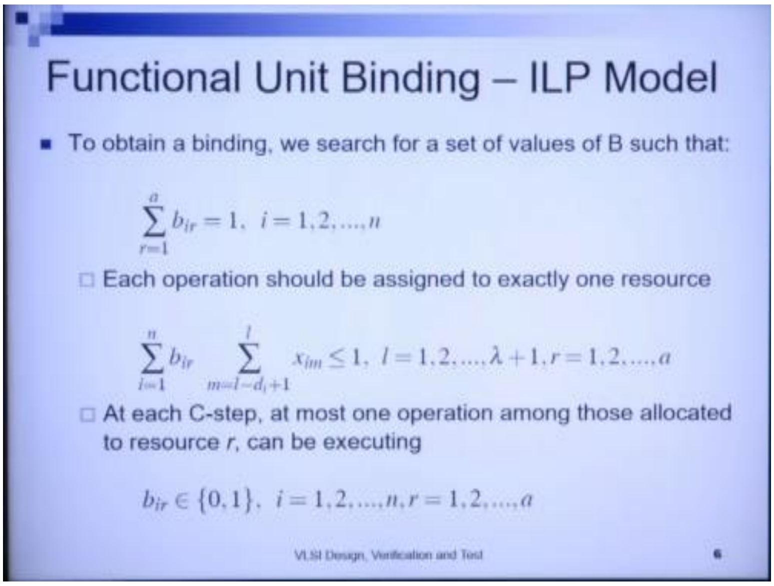
Now with this definition we first characterize resource sharing and binding problem with a simple case let us consider resource dominated circuits which have been scheduled under resource constraints so we have the resort done resource constraint scheduling during my schedule step so therefore the area is already determined by the resource usage and therefore I know the number of functional units that will be required now binding in this case binding and sharing in this case will therefore just be to refine the structural information so that the inter connection synthesis can be performed by inter connection synthesis I mean let us say subsequent allocation and binding of subsequent resources for example we said Max's buses control units etc… right.

So when we have done an actual functional unit to behavioural operation mapping then only can we do the subsequent synthesis solve the subsequent synthesis problems of our Max's buses etc in this here for simplicity we assume that all operations and functional units are of the same type let us say that so what we are doing here we will characterize the basic resource sharing and binding problem by understanding what its ILP model is like we did for the scheduling problem.

Now to start with the ILP model again here we need to define a set of decision variables so what are the decision variables here the decision variables are a set B where it contains many decision variables of type B I are now what are these indices I and R I goes from 1 to n the number of operations and r goes from 1 to dot up to a the number of resources let us say a is a bound on the number of resources and I are the operations in my operation constraints graph now equals to 1 that implies that operation VI in my operation constraint graph is bound to resource instance are now a less than n is an upper bound on the number of resources.

Now because we have already conducted scheduling my decision variables that I had for fuelling they have now become decision constants these X is have now become decision constants what do these X's tell me this X I L gives me the start times of resource I when this exile equals to 1 it means that operation VI starts at C step land that step L is given by TI right so we saw these decision variables and how these start times are obtained in the scheduling step so now they are decision consumer constants meaning these values of these constants are known to me when each operation will start at what time step is now known to me at the allocation and binding step

(Refer Slide Time: 18:02)



So with this we come to the formulation of the constraints to obtain a binding we search for a set of values of B such that that means we need to obtain a binding that means we need to or allocate the behavioural operations to the resources such that a set of constraints are met and what are these constraints the first constraint is that each operation should be assigned to exactly one resource.

How do we write this constraint for R equals to one to a summation bir that means for each operation I goes from 1 to n this has to be true for each operation now let us say for the i8 operation let us say for the first operation one b1r should be one where r goes from 1 to M that means operation one can be scheduled in only one and one exactly one of these a resources that I have only then will the submission be equal to one.

So operation one because be one I am considering can we can be assigned to at most and exactly one Operator type or resource type and hence for overt all the resources summation over all the resources this value bir will be equal to one the next constraint is that at each system at most one operation among those allocated to resource are can be executing.

So what does this tell me that for each resource are so we will take one resource at a time and for that resource we will consider for all time steps for each resource we will consider all time steps and we want to constrain we want to provide this constraint that a single resource will we should execute at most one operation zero or one at most one operation at each time step the resource can I be idle at that time step or can execute at most one operation it cannot execute multiple operations at the same type a single resource cannot execute multiple operations at the same time.

So this is what this constraint restricts and how do we write this constraint the second part of this constraint this term here this term should be familiar to you from the scheduling step what does this term tell me it tells me whether at the current time step this operation is currently under execution so summation m equal to l –di+1 to l x IM will be equal to one if operation I is currently executing at time step l and when will operation I be executing at time step l if it has started at most DI time steps earlier to L that means at El it will be executing if it has started at most D I time steps earlier than this if it has started even earlier than di time steps from L.

Then it has already finished if it has not started after di time's up then it has not started at all so and it is not executing so the at the current time step operation I is executing only if it has started at most D I time steps earlier than the current time step and this value will be one when operation I is executing at the current time set.

Now we come to the first term here what does this first term tell me I equals to1 2 and bir that means what each resource are what do I want to determine I want to find out what this term if you see in seclusion this term will give me all operations that are bound to resource are because we are doing this whole constraint for each resource at each time step.

Then if it this operation this term in seclusion this will give me all operations that are bound to resource are and when I am when we multiply this term with this term it gives me all operations that are bound to resource are and is actively executing at time step l so this gives me all operations this gives me whether and operation is active or executing at time step l and this term gives me all operations that are bound to resource are.

So the multiplication of these two terms will give me all those operations bound to resource are and are currently executing now there cannot be two operations at a given time that are bound to resource are and are both executing and hence this in this value the multiplication of these two terms has to be less than equal to 1therefore we get what we want at each see step at most one operation among those allocated to resource are can be executing.

And the third constraint says that these are decision variables are binary it can either take 0 or 1 using this ILP model we will be able to get an appropriate mapping of the resources to the operations thus with an understanding of the basic ILP model for resource sharing and binding we come to the end of module 1 of lecture 6.