

INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

NPTEL

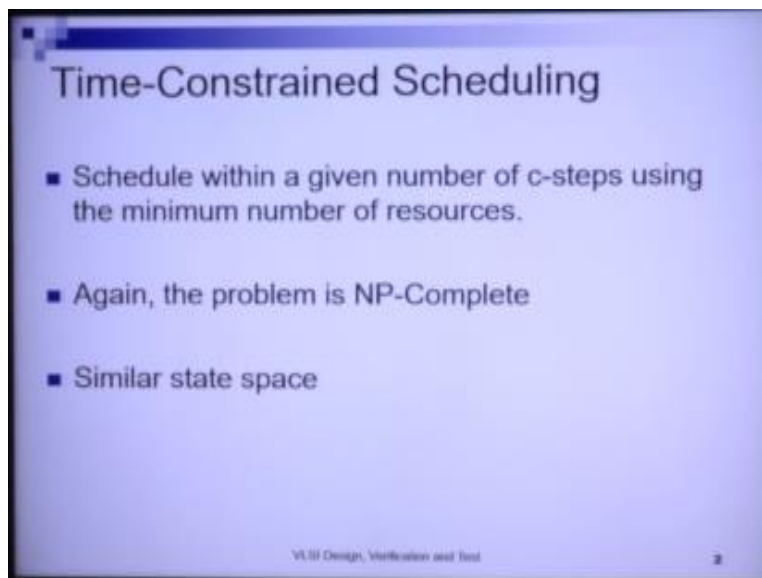
NPTEL ONLINE CERTIFICATION COURSE
An Initiative of MHRD

VLSI Design, Verification & Test

Dr. Arnab Sarkar
Department of CSE
IIT Guwahati

In today's lecture we will look at time constraint scheduling so time constraints scheduling.

(Refer Slide Time: 00:32)



Is the problem of allocating c-steps to operations using a minimum number of resources given a latency constraint, so we will be given that we must schedule we must complete the scheduling within a given number of c-steps and within that given number of c-steps with that upper bound on the given number of c-steps I must try to minimize the amount of resources that I will use. So this problem like the resource constraint scheduling problem is an NP-complete problem and has a similar state space.

(Refer Slide Time: 01:12)

The ILP Model

- Minimum –resource scheduling problem under latency constraints
- Objective: Minimize $(\sum_{k=1}^{n_r} cost_k \cdot a_k)$ such that :
 - $cost_k$: Area cost for resource type k
 - a_k : Number of resources of type k ;
 - a_k is now an unknown auxiliary variable
- Latency Constraints: $\sum_{l \in I_i} l \cdot x_{ij} \leq \lambda + 1$
 - λ represents the upper bound on schedule latency

VLDD Design, Verification and Test 3

So before as we did for the resource constraint scheduling problem we will look at the ILP model, so the ILP model for the latency constraint scheduling problem is very similar to that of the resource constraint scheduling problem. The main difference being the objective here differs, so what is the objective here we want given a bound on the time within which to schedule I need to minimize the resource that is consumed that means the area that is consumed by the circuit.

I want to minimize the area that is consumed by the circuit given a limit on the time that I can use to schedule the operations in my tag. So how do we specify this objective minimize k equals to 1 to n our cost k into a_k so if this is what does this mean your cost k is the area cost for resource of type k and a_k denotes the number of resources of type k , so what do we want to minimize here we want to given the area that a certain resource a certain functional unit will take means I know before the scheduling for each type of resource I am given the type of resources.

Now for each given type of resource I know the total amount of area that is consumed by 1 instance of the resource that is known to me. However through scheduling I need to find out how many resources of each type should I use so that I will be able to schedule all the operations in my tag in the given time constraint and while minimizing the total resource consumed by total

area consumed by this set of resources, so that is why for resource suppose I have two resources multiplier and adder then let us say a_1 is adder and a_2 is multiplier then cos_1 is the cos area cos of the adder and cos_2 is the area cos of the multiplier, so an important thing to note here is that unlike the resource constraint scheduling problem a_k is not known that means the number of resources of type k is not known it is now an unknown auxiliary variable and that has to be determined by appropriately scheduling the operations .

Also in this problem we need a latency constraint a specific latency constraint and let us say that latency constraint given to us is λ . Now in the previous 1 we also had an upper bound of the latency constant λ but we wanted to minimize the time within which to schedule all operations given the number and types of resources here we also have a time constraint λ but here we want to minimize the so that max within this λ I can schedule all my operations.

So what do I want what is the latency how is the latency constraint specified it is specified by saying that here it will here I have written a bit wrong in the sense that this i variable will be n , so what I want is that the init variable this whole expression tells me the start time of the n th operation the n th operation that means the sink node I am saying that the sink node should be scheduled at most within $\lambda+1$ so for all the other nodes sink node is a dummy node so for all the actual operations that I have that must be scheduled within λ and the start time of the sink node should be less than or equal to $\lambda+1$.

So here again I am telling that there is a mistake in the expression here i should be replaced by n the sink node so what I want to say again I want to specify that the start time of the sink node should be less than my time constraint plus 1, so the overall ILP formulation for the time constraint scheduling problem becomes minimize.

(Refer Slide Time: 06:07)

The ILP Model

Minimize $(\sum_{k=1}^{n_r} cost_k \cdot a_k)$ such that :

$\sum_{l=t_i}^{t_j} x_{il} = 1$ ← Unique Start time constraints

$\sum_{l=t_i}^{t_j} l \cdot x_{il} \geq \sum_{l=t_j}^{t_i} l \cdot x_{jl} + d_j \quad : (v_j, v_i) \in E$ ← Dependency constraints

$\sum_{i: r(v_i)=k} \sum_{m=l-d_i+1}^l x_{im} \leq a_k$ ← Resource constraints

$\sum_{l=t_i}^{t_j} l \cdot x_{il} \leq \lambda + 1$ ← Latency constraint

$x_{il} \in \{0, 1\}$

VLSI Design, Verification and Test 4

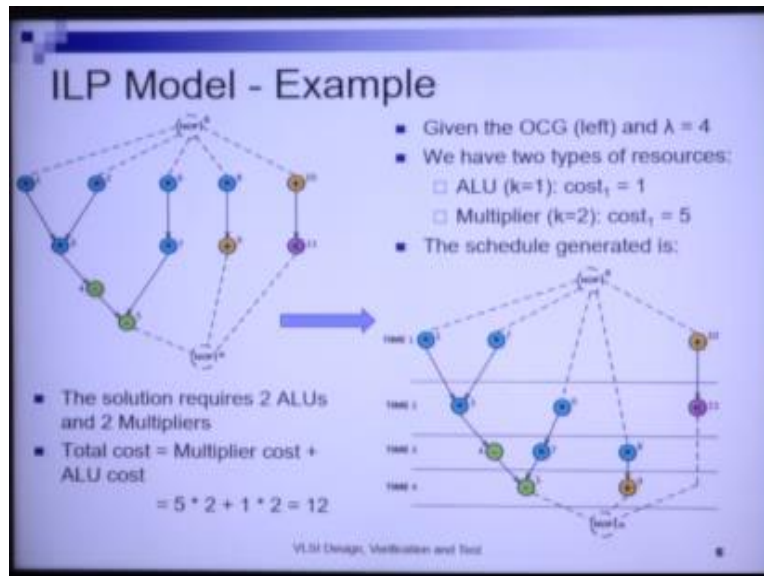
The objective function such that the constraints that are specified will go through the constraints just brush up brush them up very quickly the objective function again is what objective function attains to minimize the total area of the circuit why such that the operations can be scheduled within the time constraint λ . Nowhere the first constraint here specifies the unique start times of all nodes as we have seen in the last lecture this one the second constraint gives me the dependency constraints that means if I have two nodes with edges dependency or precedence constraint edges between them.

Then 1 must be scheduled after d_j times the other has complete after d_j times the other has started so the start time of the dependent operation should be d_j times sorry, after study after d_j amount of time the first operation where d_j is the delay or execution time of the first operation. Now we also have resource constraints we said that for each resource type at each time step I can only use a maximum number of resources I can only use a maximum number of a_k resources of type k so that that is how we specified resource constraints.

But again here a_k is not known and is an outcome of the scheduling problem it is an auxiliary variable here and then we specify the latency constraint which is specified by the start time of the

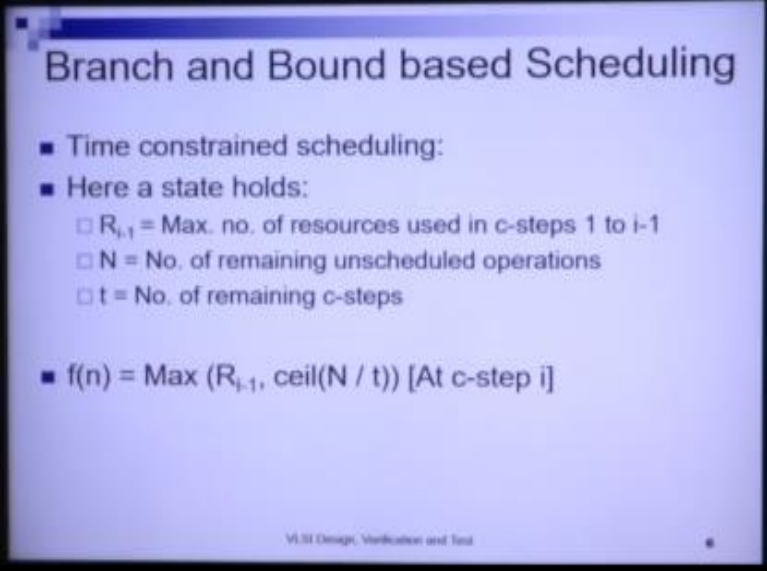
nth or the or the start time of the sink node, right so the sink node must start at most within $\lambda+1$ time steps.

(Refer Slide Time: 08:06)



Given this formulation we will now take a small example, so how does this formulation proceed given this operation constraints graph on the left let us say we use the same time constraint $\lambda=4$ we know the solution but we will go through the steps here while doing time constraint scheduling. So we are given the operation constraint graph and we are given the given the constraint on the time we are also given the types of resources so for k equals to 1.

(Refer Slide Time: 08:56)



Branch and Bound based Scheduling

- Time constrained scheduling:
- Here a state holds:
 - R_{i-1} = Max. no. of resources used in c-steps 1 to $i-1$
 - N = No. of remaining unscheduled operations
 - t = No. of remaining c-steps
- $f(n) = \text{Max} (R_{i-1}, \text{ceil}(N / t))$ [At c-step i]

VLSI Design, Verification and Test

ALU I have a cost I have a cost of 1, for k equals to 2 multiplied i have a cost of 5 I have a cost of 5, so what does it mean it says that it says that ALU has an area cost of 1 and multiplier has an area cost of 2 here this is not cost 1 but cost 2 so $\cos 2$ is 5 that means the multiplier unit functional unit or resource consumes an area of five on the circuit and the ALU which is k equals to 1 has a cost of 1 meaning that the ALU consumes an area of 1 on the circuit.

Now given this we know that to schedule this we need at least two ALU's and two multipliers from the resource constraint scheduling problem, so this is what we did not go through but this is what the ILP will exactly tell me the time constraint scheduling ILP will also give me the same solution that to share deal with in for time steps you need at least two ALUs and two multipliers and that is the least cost solution that you have in terms of area required so what is the cost, so let us say if I require two ALUS's and two multipliers as I have said the total cost will be the total multiplier cost plus the total ALU cost.

So what will be the area cost total area cost of the circuit so five is the cost of 1 instance of the multiplier I have two multipliers so the total cost is 10 and 1 is the cost area cost of the ALU I

have two L use so 2 is the total ALU cost, so the total area cost of the circuit becomes 12, so with this understanding of the ILP solution we probably now briefly see how the branch and bound solution differs for the time constraint scheduling problem with respect to the resource constraint scheduling problem. Now in the time constraint scheduling problem at any given state of the state space what do we what do we store we stored the maximum number of resources that I have consumed in steps 1 through $i-1$ so we take the same simplistic assumption that I have one type of resource three instances of that type of resource and everybody we are consumed we are assuming has a d_j of 1.

So what will be the max resources that means at each time step I will require some number of resources okay, here I need to minimize the number of resources or is not given to me here I want to minimize the number of resources. Now at any given state i have started from the root and have come to this state at any given state the way in which the state space will be expanded will be a bit different here from for the time constraint scheduling compared to the example that we took for the resource constraint scheduling why, because here we will know more exhaustively allocate all resources.

We will know more exactly allocate all resources to all the time steps we want to minimize resources rather the number of resources is not given to me our choices will be allocating each operation to a distinct resource I may allocate an operation to a resource I may not have located operation to a resource and we although the other things what we discussed for the for the branch and bound remains same that I need if all my if all my operations at a particular time step gets exhausted I need to increment time to get back any resource that I have, right.

So at any point in time the branch this branch and bound here we will keep an account of the maximum number of resources that has been used in see steps 1 through $i-1$ why because if that many resources have already been used from the initial state to the current state that when resources will please be required because we have already used in a certain time steps or three resources in parallel let us say.

So three resources will be required in the in the solution what is the solution we again set apart from the root to the leaf in the state space tree is a solution is an enumeration of that solution, so in that part if the maximum number of resources that has been consumed at a time step is three there is no way that I can execute that I can produce a schedule with this path that consumes less than three resources, so the answer for this part will be three resources so what is the basically the answer in the time constraint scheduling problem what is the maximum number of resources that has been used up in any time step in that particular solution from the root to the leaf, right so that will give me a solution.

Now n at that state, n gives me the number of remaining unscheduled operations and t is the remaining number of c -steps. Now for the resource constraint scheduling problem we did not know we wanted to minimize the number of c steps in which I want to shade in here I am given the number of c steps that I have so I have currently already used a certain number of time steps and I have a certain number of time steps within which to schedule my remaining operations and my basic objective is to minimize the consumption of resources such that all my operations will actually be scheduled in those remaining number of steps that starting from this state.

So what is my pruning function here FN by pruning function here I will first start with a very crude cooling pruning function a very crude pruning function here would be that the maximum of R_{i-1} which means that the maximum of the number of resources that I have already used in any of that the prior time steps between $i=1$ to $i=y-1$ that means before this term before the current time step what is the maximum number of resources in any given time step that i have used.

And a lower bound on the minimum number of resources that may that I may need to use at any given time step for the remaining time steps, so I have a remaining number of time steps in any of these time steps what is the lower bound of the maximum number of resources that I may need now a max of R_{i-1} and $\text{ceil}(N/t)$ so $\text{ceil}(N/t)$ it gives me what N is the remaining number of operations t is the remaining number of time steps.

So N/t is the minimum number of resources that I may need at any time step given that I can use the maximum there are no dependency constraints, given that there are no dependency constraints N/t is the minimum number of resources that I may need at all time steps in the remaining t time steps that I have. Now $R_{i-1}, \text{ceil}(N/t)$ max of this gives me the lower bound for my current state. So why do we use this event, we use this event to compare with the current best solution that I have.

So I have already traverse through another path of the state space and I am already found a current solution. Now we will compare at this state with the current solution, if the current solution the current best solution tells me that it can schedule with a lower number of resources within the time bound then my solution a lower number of resources than the lower bound that is produced by at this state so at this state f_n gives me a lower bound of the minimum number of resources lower bound of the number of resources that I will require at any time step.

Now when I am comparing with the current best solution what I am getting is that when is the current best solution be always better than the lower bound when will it be when the current best solution has a lower value, the current best solution will have a lower value meaning that in that solution the maximum number of resources that I may require to use in parallel at any time step is lower than the minimum number of resources that I require for this solution.

So at this state I can prune the rest of the sub-tree which means that I do not need to explicitly expand the rest of the sub-tree and can look into other parts of the solution of the state space to find a better solution. Now $\text{ceil}(N/t)$ is a crude solution we can still use better and better solutions but I leave it to the student to find these solutions and we may discuss outside the scope of these lectures in the assignments.

Centre For Educational Technology

IIT Guwahati

Production

Head CET

Prof. Sunil Khijwania

CET Production Team

Bikask Jyoti Nath

CS Bhaskar Bora

Dibyajyoti Lahkar

Kallal Barua

Kaushik Kr. Sarma

Queen Barman

Rekha Hazarika

CET Administrative Team

Susanta Sarma

Swapan Debnath