INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

NPTEL

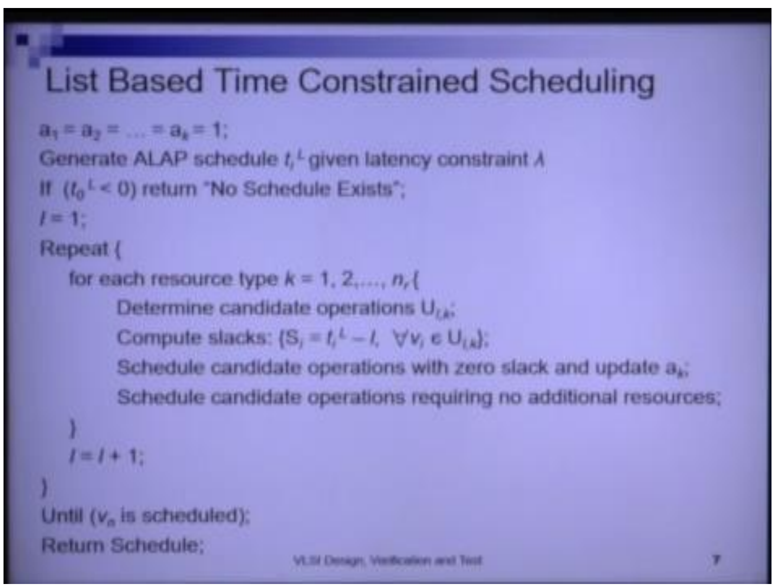
NPTEL ONLINE CERTIFICATION COURSE
An Initiative of MHRD

VLSI Design, Verification & Test

Dr. Arnab Sarkar
Department of CSE
IIT Guwahati

You hello welcome to module 2 of lecture 5 in the last module we looked at time constraint scheduling and optimal algorithms for doing time constraint scheduling in this lecture in this module we will look at heuristic strategies for doing time constraint scheduling .

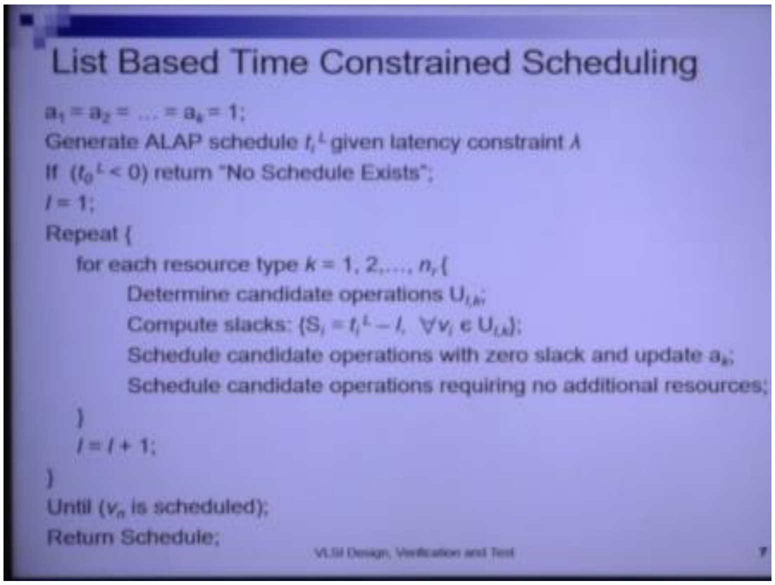
(Refer Slide Time: 0:50)



So just to reiterate what is the problem of time constraints scheduling it is the problem of scheduling the operations in an operation constraint graph within a given time bound latency known as λ using the minimum area and how do we measure this minimum area in terms of the resources it uses last day we said that each resource type has a cost which is a measure of the area it takes on the floor of the that it is expect to take expected to take on the floor of the chip.

And when it is known that what is the area of each type of resource on the floor of the chip the problem is to minimize the summation over cost of the resource into number of resources so we looked at the optimal version of those problems of that problem and today we will look at heuristic solution strategies.

(Refer Slide Time: 0:1:53)



The first heuristic solution strategy is a list based strategy so when we talked about list based scheduling in the resource constraint sheet in the resource constraint scheduling problem we said that the basic strategy is this we progress time step by time step in each time step there are a set of ready to schedule operations and these operations have a defined priority on them and for each type of resource we schedule the highest priority resources of each type right so this was the basic principle of resource constraints so the same principle will be followed here however there the priority function for resource constraint scheduling was that it was the weight of the longest path of the critical path from a given node to the sink node.

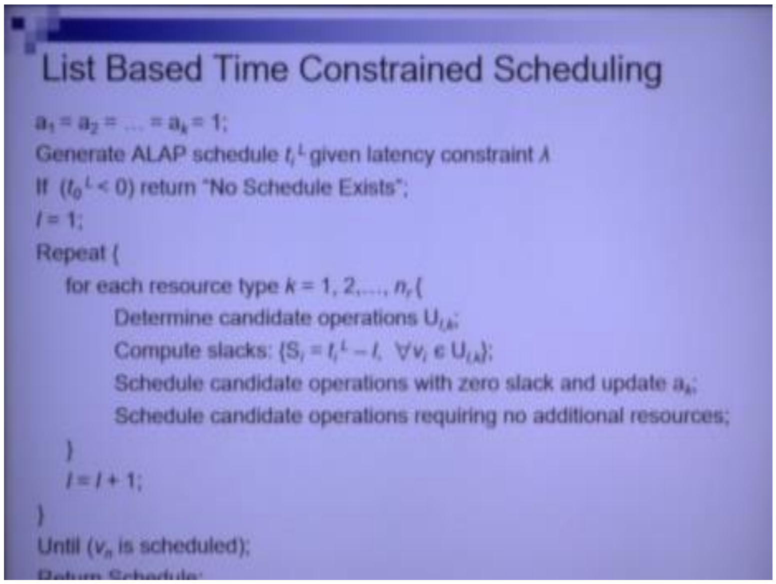
Now here because it was a resource constraint scheduling problem we had to minimize time given abound on resources now here the priority function will be slightly different priority function here is defined by the computation slack available to a node with respect to the latest valid start time that it has so how do we find the latest valid start times we find the latest valid star start time using the elapsed idyllic method logic so given the elaborate ALAP methodology and the current given the time step given by the ALAP methodology and the current time step we can find the computation slack as the difference of the AALAP time - the current ,current time step now so how does this scheduling algorithm proceed firstly for each type of resources we assume that we have only one unit of that resource available.

So we want to minimize resource usage we only have 1 unit of that resource available now we do the ALAP schedule to generate the IL values given latency constraint λ here we are given a latency constraint bound λ and we find the ALAP based on that now obviously if the elapsed time that is the latest time that it that it can be scheduled in that, that the first node of the operation constraints graph will be scheduled in is less than zero then no solution exists how can this be less than zero the latency bound is so constrained that even.

If we have infinite resources it even if we give infinite resources we cannot schedule the critical path in time and hence the start time of the first node of is less than zero in that case no solution obviously exists otherwise again as we said the scheduling progresses time step by time steps so =1 the first time step and similar to the previous list scheduling algorithm the algorithm proceeds for each resource type for each resource type k +1.. Up to 1 similarly we determine the candidate operations which means the radio operations that are available at this time step and then for each of these and unscheduled and radio operations at this time step we calculate S_I what does S_I give me S_I gives me the slack.

That is what we were looking at TIL -L is the slack at this point in time and then what we do we schedule candidate operations with 0 slack so somebody whose slack has reached 0 they must be scheduled for that if we need extra resources we have to give extra resources because if you do not fit you this operations at that time step then what will happen then we cannot meet the time bound we will miss the deadline and then if we then based on after the 0 slack we have found operations with 0 slack and we find whatever number of resources that is required at the time step based on that we update $a_{,k.}$

(Refer Slide Time: 0:6:31)



So a_k is the number of resources of type care and we are handling resources of type care in this iteration after that if we have operations left so you when we have updated a_k for all subsequent time steps the minimum number of resources available to me is a_k right so after we have scheduled the 0 slack operations and let us say not more than a_k resources that means less than a a_k resources have been required for that and we have some residual resources that can change more operations we can use this residual remaining resources because anyway .

We are not enhancing the number of resources required in the schedule so you see have such remaining resources then schedule candidate operations requiring no additional resources after doing this we increment time and we repeat until the schedule is done so again what are we seeing here for each resource type k=1 2 .. up to n we let us take we will take the first resource type we determine the candidate operations for this resource type that are unscheduled and ready for these unscheduled and radio operations we find out the maximum slack what is slack is a measure of the urgency of this operation at this current in time who are the highest adult operations.

At any given time those with the least slacks so if we have if there are candidate operations with slack 0 they must be scheduled at that time step otherwise the deadline will be missed if we have such operations then we update then we shall all those operations at that time step now if there is a current value of a_k that we are having right now if the number of operations with 0 slack are more a_k that means all those these operations which I must should you like this time step will require more than the number of resources that I have already allocated I have to increase the number of resources I have so we have to update a_k .

We have to increase a_k starts with 1and and depending on the number of operations that I have to schedule at a time step a_k increases now however at all steps we will not be a will not have that many operations as a_k suppose in a prior time step we had many operations with 0 slack and hence we had to increment a_k but in the next subsequent time step there are not that many operations with slack zero so the number of operations with slag zero is lower than what was required in a prior time step so therefore the value of a_k will be higher than the number of operations with 0 slack.

Now we first allocate all these 0 slack operations on to the resources but I will have a set of remaining resources at this time step these time step these resources can be allocated now to other operations with slack higher than 0 so sheer dual operations requiring no additional time steps obviously we will we will start they will not be 0 but we will be scheduling operations with the lowest available positive slack on these resources now when we have scheduled or all the resources in operations on all the resources at this current time step we increment time we increment time and again a set of operations become available to me untitled opposite another set of unscheduled operations become available to me.

And we are then compute the slack because the slack may have may have varied suppose I have a set of unscheduled operations that have not actually allocated in the current time step for all those operations the slacks in the next time step will reduce because the value of l has increased so TI l - n will TI capital L minus L will decrease so we have to recomputed slacks again we scheduled operations in with 0 slack and update a_k and then again scheduled operations requiring no additional slack this continues until all operations has been scheduling the graph so this is the list scheduling based heuristic strategy for time constraint scheduling with this we come to the end of module 2 of lecture 5 .