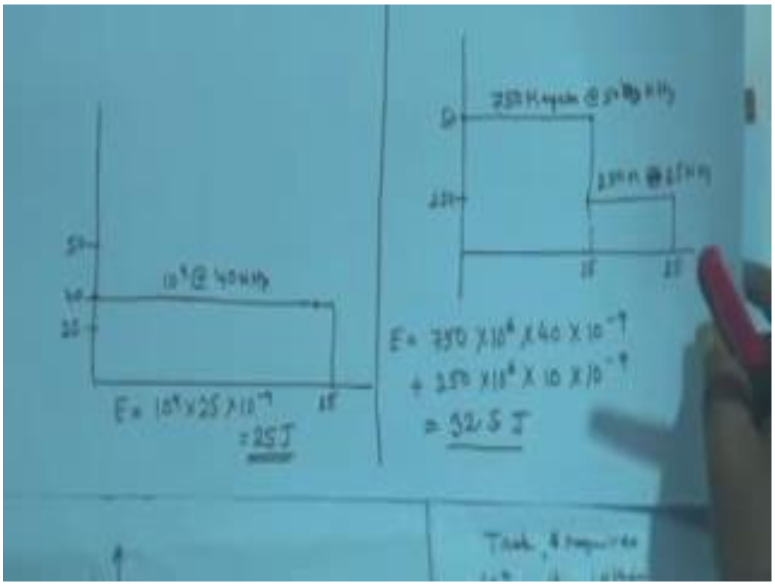
Embedded Systems Design
Prof. Anupam Basu
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 21
Power Aware Embedded System – II

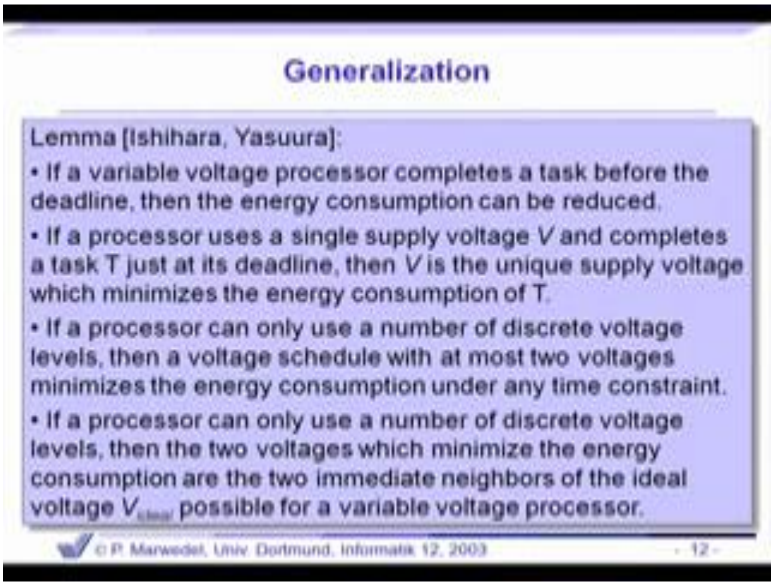
In the earlier lecture we have seen three cases right that the same task which requires 10 to the power 10 raise 9 cycles.

(Refer Slide Time: 00:37)



If I run it at 5 volt all through; I consume an energy of 40 joule and the same task; if I run in two slots like 750 million cycles at 5 volt and 250 million cycles at 2.5 then I will get 32.5 joule. On the other hand, if I had run the whole thing at 4 volt then I would have consumed 25 joule. Now my question is that we are getting the minimum energy consumption in this case right.

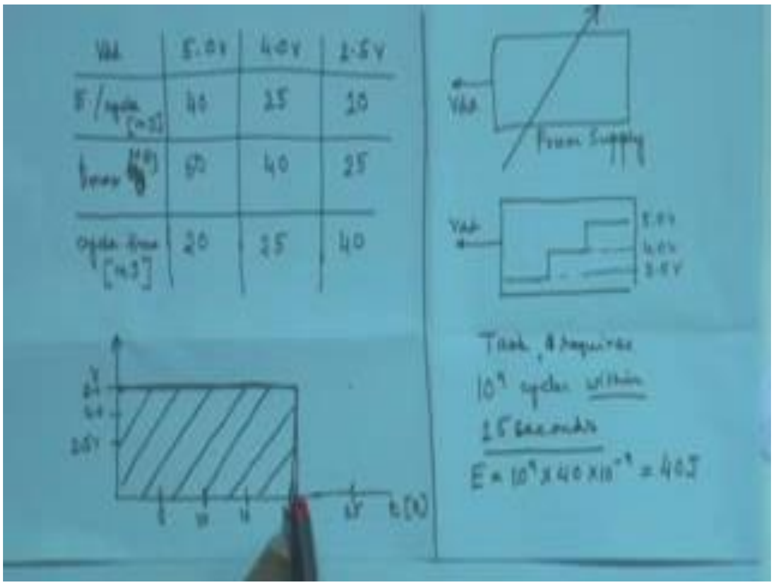
(Refer Slide Time: 01:32)



So, therefore this observation can be generalized in the form, it is of a Lemma by Ishihara and Yasuura; that if a variable voltage processor completes the task before the deadline then energy consumption can be reduced. For example, if I had run this one at completely run it at 5 volt here and it has finished before the deadline then of course I can reduce it.

Now, here when we say a variable voltage processor; a variable voltage processor means a processor which can be continually whose voltage can be monitored. The second point is that if a processor uses a single supply voltage v and completes a task just at its deadline then v is the unique supply voltage which minimizes the energy consumption.

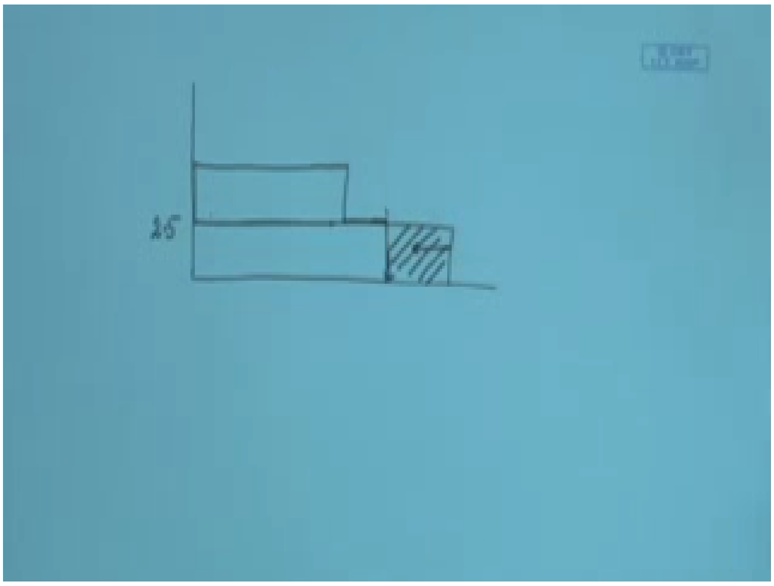
(Refer Slide Time: 02:47)



Again coming back to the earlier example say here, if my deadline was 20 seconds then 5 volt with 5 volt I have reached that then I do not have any other way of reducing this volt clear that is the best that I can have. The next point is if a processor can only use a number of discrete voltage levels; our case we have 5, 4 and 2.5; if a processor can only use a number of discrete voltage levels then the voltage schedule with at most two voltages minimizes the energy consumption under any time constraint; at most I will need two voltage levels.

The reason is simple I mean if I have got discrete and I can minimize say if I take it to the lowest voltage level then it will minimize, but at this what can happen, it will overshoot the deadline then I will have to select another voltage level which is higher than this so that the deadline is just shifted.

(Refer Slide Time: 04:08)

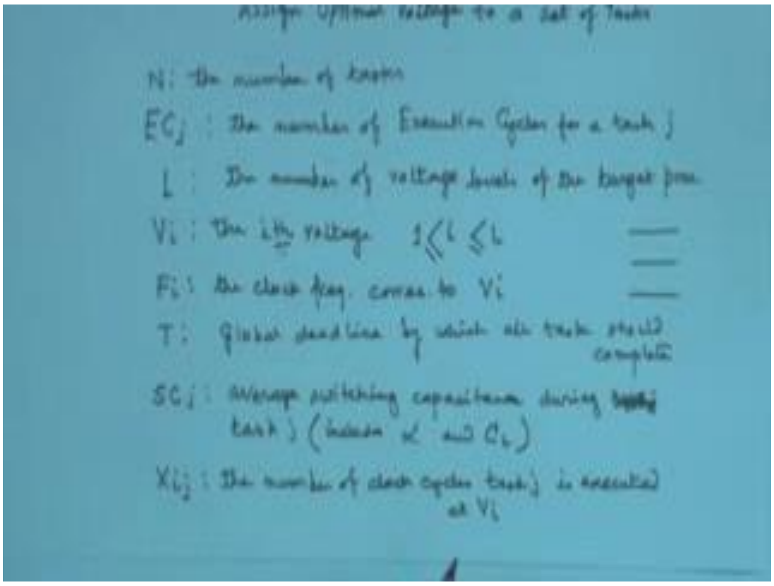


So, things will if I select a voltage; minimum voltage and I find that the deadline was here and I have crossed the deadline but I have done it with a minimum energy then for this reasons, so that I can shift this part on this side; I will have to select some parts; some parts we will have to say this part can be done at a higher level minimally; so that I can complete the (Refer Time: 04:45) within this time. So, it is a shift right that I can do, so with two levels I can manage.

The last one is if a processor can only use a number of discrete voltage levels, then the two voltages which minimize the energy consumption are the two immediate neighbors of an ideal voltage for a variable voltage processor; what is an ideal? Ideal is where if I had continually could do that and it will be minimized for the ideal one then the voltages in the discrete scenario that will be achieve it are the two neighbors of this ideal voltage; closest neighbors.

So, this is the generalization that we get from the example or such examples that we have seen. But how will we solve this problem? Here we have just shown some examples and tried to solve it; show it by trial and error we could do but in general how can we do that if there be a case of multiple tasks. Here we could do that by tweaking the things because there was only one task, if there are multiple tasks which have to this completed each having it is own deadline then what would be my allocation of voltage to the different tasks? How would I scale the voltage for the different tasks?

(Refer Slide Time: 06:32)



The model that we are trying to do is we want to assign optimal voltage to a multiple task and we will take a linear programming model for that; our job is assign optimal voltage to a set of tasks. Now let us assume I would suggest that you note down so that you can check it later. N is the number of tasks, ECj is the number of execution cycles for a task j, now here there is an issue; how many execution cycles a task will take? I mean it cannot be always known deterministically because the task can take different paths depending on different conditions. So, what we actually analyze is the worst case execution time, the longest path that it can take.

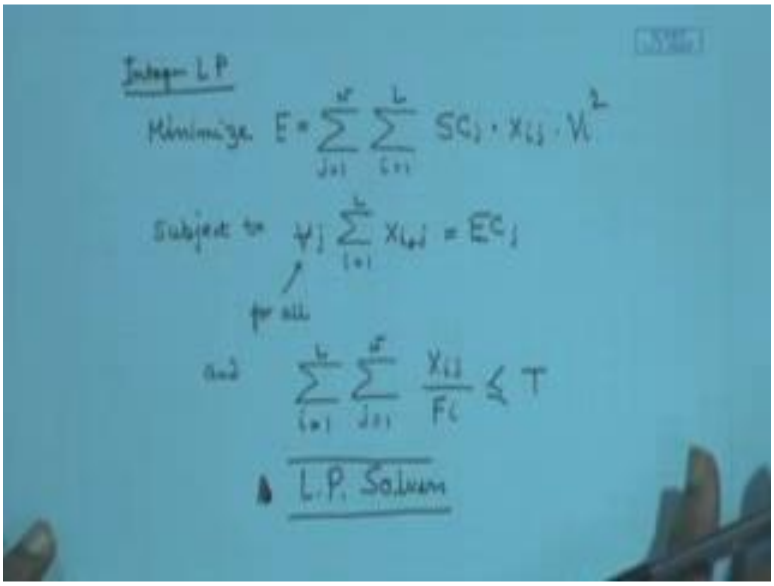
L is the number of voltage levels of the target processor, the target processor has got L number of voltage levels V i is the ith voltage. So, i will vary between 1 to L because I have got L voltage levels, there are L voltage levels 5 volt, 4 volt, 3 volt, 2.5 volt. So, any one of those can be taken corresponding to that voltage F i is the clock frequency corresponding to V i and T is the global deadline by which all the tasks should complete by time T.

There are still two parameters one is SCj; this is the average switching capacitance during task j. So, SCj is actually comprising of includes what; includes C L alpha and the capacitance. You see, if I again go back to this earlier this diagram then the number of switching that is taking place is taking across this capacitance. So totally with that; that is the actual capacitance that will have with respect to; so much switching as alpha increases that alpha C L component; so, that is alpha C L component and Xij; number of clock cycles task j for which the task j is executed at voltage V i.

So, say for example if I take this example that we had done, about this task has been run at voltage 5 volt for so many cycles 750 into 10 to the power 6 cycles. So, that is Xij alright and SCj is the average capacitance, so let us try to understand this because when I move to the L P formulation, we will have to keep a reference to that. So, I would request all of you to note it down. So, once again I repeat N is the number of tasks and ECj is the number of execution cycles for tasks. For our earlier example, the only task we had it is ECj was 10 to the power 9 cycles; L is the number of voltage levels. For example, we had three voltage levels in the earlier case, V i is the particular voltage level at which you are running, F i is the clock frequency corresponding to voltage level; so we found for 5 volt we are getting 50 mega hertz like that, T is the global deadline.

Our global deadline in the earlier example was 25 seconds; SCj is the average switching capacitance during the task j that is taking place and the number of clock cycles that at a particular voltage level a task executes for; is given by Xij, so our task is to minimize the energy.

(Refer Slide Time: 14:19)



Now, so we can say our in our L P model; the linear programming model, we want to minimize energy which is given by j equal to 1 to N that is for all the tasks and i equal to 1 to L; that is for all the voltage levels, I must have SCj times Xij; V was the voltage, V i square that is what should be minimize SCj is what? SCj is alpha C L; Xij is the number of clock cycles, so that gives me the number of clock cycles there and V i square is this.

Now this we will have to minimize subject to the constraint that for all j; this notation means for all j; i is equal to 1 to L; Xij should be ECj because ECj was the number of execution cycles for the task and for all the different; I am sorry this is not Xi, there is no comma here; this Xij is the same as Xij here the number of clock cycles. So, that summed over for all task should be ECj; that is by definition and another constraint is sigma for i equal to 1 to L; that means, for all the voltage levels and j is equal to 1 to N for all the tasks; Xij by the frequency corresponding to a particular V i that should be less than or equal to the T, where T is the global time by which.

So, these are the constraints and this is each of these constraints are 1, 0 type of constraints. So, these are this is an integer programming model or integer L P or I L P you call it; model of solving this. Now if design this thing, I can formulate this and we have got several tools which are the L P solvers like LP solve is 1 and we will find a number of tools which are L P, L P solvers where we can feed these constraints and we can get the solution for multiple tasks.

So, major challenge for the engineers right now is to formulate this properly. So, here is an example where for this case that we have discussed in the earlier lecture also that we extended that, that was for only one task but by extending it to more than one task, we can formulate this integer programming problem and run it and solve it.

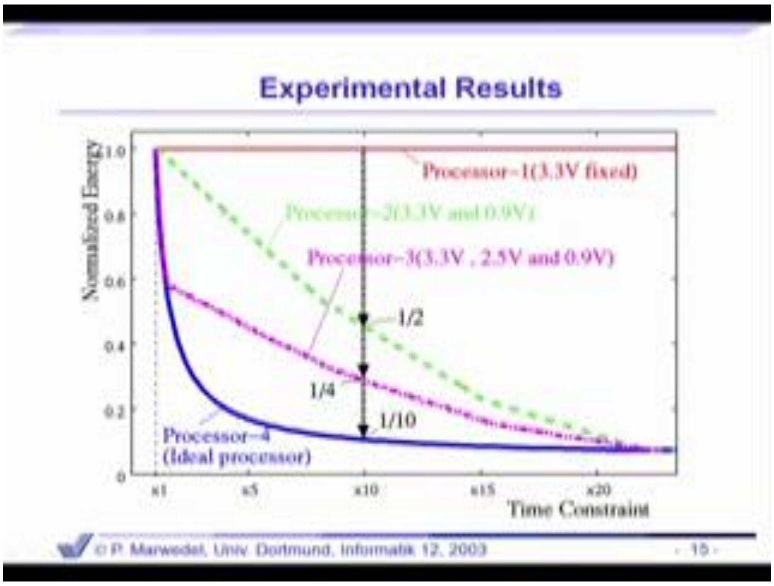
Student: Which is the last thing Sir?

This one L P solver.

Student: (Refer Time: 18:55).

This Xij is or the number of clock cycles, one particular task is running at V I, so one task j may be running at different i's, different voltages. For each of the i's there is a frequency, so it is consuming sometime for each of them. So, I will take the summation of that for all the voltages, a particular task is going through, so for each of the voltages it is expending sometime. So, I am summing that up and that I am doing for all the tasks and that should be of less than the absolute time; the global deadline which has been given by which all the tasks should be finished. So, for each of the tasks I am finding out and then adding them.

(Refer Slide Time: 19:53)



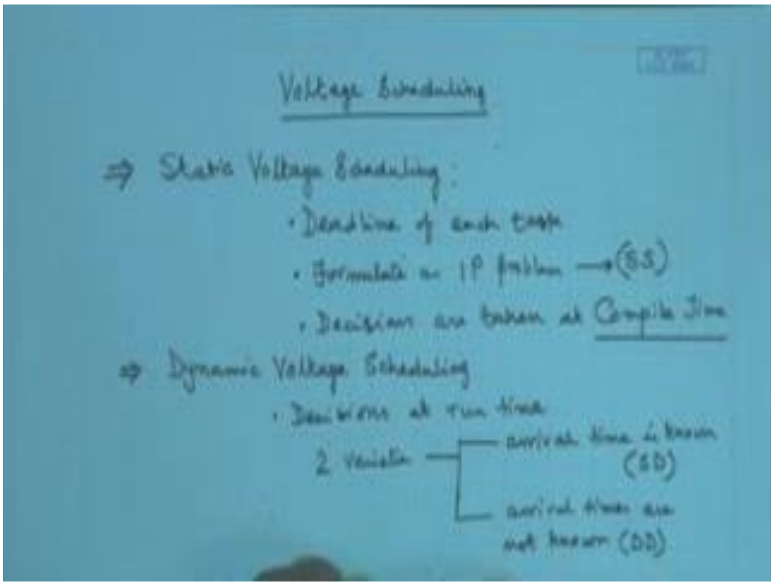
So, now here is a typical experimental result you can see this that say for and remember the earlier generalizations that we have seen. So, if we run at a processor like a typical fixed processor, the typical microcontrollers or we know or whatever you are using; you have got 3.35 volt fixed and you have got the normalized energy to be this but if I had 3.3 and 0.9 volt then it comes down as the time constraint increases, I am getting less and less energy. As I go for higher p level voltages, it still comes down and processor four is being shown as the ideal processor where there is no discretization; it is continuous I could have got even more.

If the time constraint is tight; that means, I move to the left, then more energy is consumed because I have to push more cycles at higher voltage. As the time constraint is loosened then I have got the luxury of running different parts at lower voltage, so the energy consumed consumption comes down. So, this is a typical representative curve to show how the normalized energy changes with time constraint.

So quickly summarizing what we have done till now; what we have done till now? We have seen the importance of power and energy that both the things we have to look into. We have seen that the actual players in power consumption are the switching alpha part, the load capacitance, the supply voltage and there is a contradicting or conflicting requirement that as I reduce the Vdd, the delay will decrease of course if I go but the frequency increases; I am sorry I just told the reverse; as I reduce the Vdd; the frequency also gets reduced, therefore, the thing will be a little slower.

But if I reduce Vdd, I can have by the power equation I will be consuming less power. So, we were trying to see that if I have a processor where I have got the opportunity of having multiple voltages then different segments of my execution whether I can scale to different voltages. So, that the power consumption can be energy consumption can be minimized and at the same time, I will be adhering to my deadline that I cannot escape. So, accordingly I can to do that and we have seen it through some examples; for a single task can for 2 or 3 voltage levels given, you can very well hand compute that but whenever it comes with multiple tasks and number of voltage levels then their contradicting requirements so that optimization can be formed, optimization problem can be formed as an I P problem and it can be solved it with some existing solvers.

(Refer Slide Time: 24:05)



Next we will talk about the topic which is voltage scheduling; now what we are doing is voltage scaling is nothing but voltage scheduling. Now what have we done, we have seen in this way when we are optimizing this, whenever we were optimizing this in this way; using this or using this L P or using our hand calculation; any of these, what we were doing is we were just having the voltages assigned apriori, we are assigning the voltage for this many cycles at 5 volt, this many cycles at 4 volt, this many cycles at 2.5 volt.

Even if that is the case then who will instruct that now the machine that now runs at this voltage, now run at this voltage. Some controller or manager will have to do that and that is none other than the operating system. So, the operating system we know schedules tasks with time and here we find that there is another task that is coming for the operating system. So, we have got two types of voltage scheduling algorithms; one is static voltage scheduling where we know the deadline of a task.

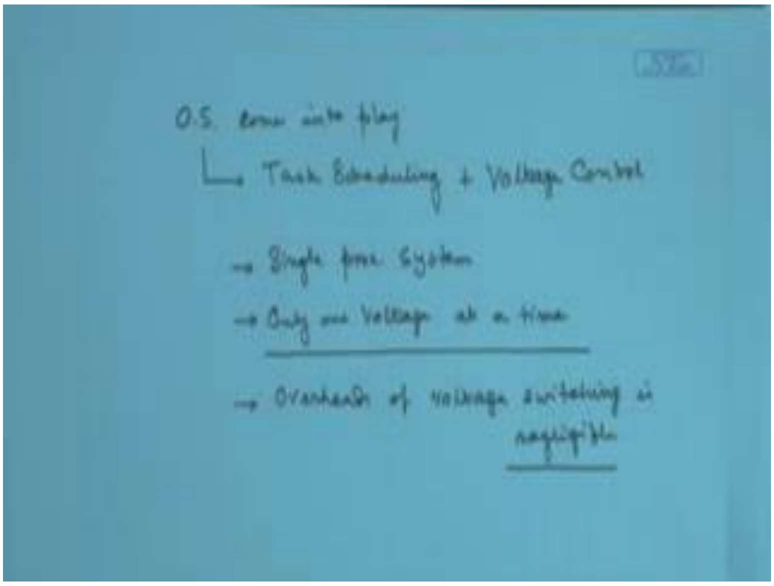
By the way I have assumed that all of you understand what is meant by a deadline. Deadline is a time that is given with respect to every task by which the task must complete; that is the deadline and if that is the case then we can formulate as I P problem and that will give us as static scheduling algorithm. The schedules are fixed and the voltages are fixed and the decisions are taken at compile time, whenever the task had been compiled I had all the information and so at that time point of time I decided; this is static voltage scheduling that we are talking about.

The other obviously is a dynamic voltage scheduling, where we are taking the decisions at runtime and there are two varieties of this. One is arrival time of the tasks is known apriori, I know the arrival time of the tasks beforehand alright; that sort of algorithm is known as the SD algorithm, where the arrival times are known statically known apriori but I am doing dynamic voltage scheduling, I am not doing at the compile time. Another is arrival times are not known, things can be asynchronous and can come at any point of time that is the DD algorithm.

So we have got two varieties; one thing is that I know a priori, I know the deadlines, I know the voltage levels available to me, I know the deadlines as well as all the other information. So, I can decide on that either it will be a single task by hand tweaking or by forming an integer L P and solving it, I can formulate and find out the schedule. The other thing is that it is at runtime, we will have to take decisions at runtime because I do not know everything before that.

The arrival time is known but other things like the deadlines and everything are not known; as it comes I will have to adapt to that. So, based on this we have got this dynamic voltage control therefore, in this case whenever I am trying to do this scheduling, I have to do this control using the operating system, so now the operating system comes into play.

(Refer Slide Time: 30:02)

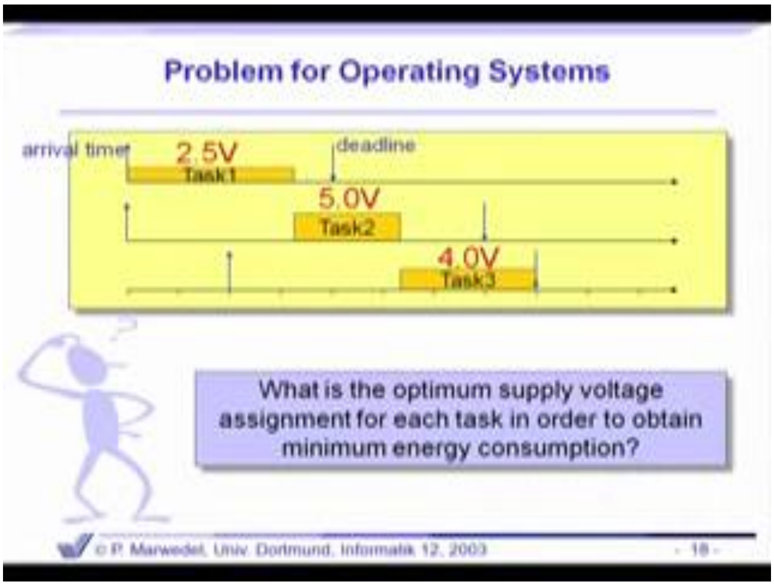


So the ways we will have to not only to do the task scheduling that we knew that all of you have learnt in the operating scheduling class; round robin, shortest job, first come first served, etcetera, etcetera. Along with that we have to do the voltage control also, both of them are required in the same time. So if we take the target as a single processor system, we are considering a single processor system here; not right now looking at the multiple systems and we are assuming that only the operating system can issue the voltage change commands, only all these things are being given by the operating system, we are also assuming that only one voltage can be applied at a time.

Please note what is the significance of this line, only one voltage at a time; I had shown another example where another approach of solving this volt; the energy problem is to compartmentalize the task into different parts and apply different voltages to different segments. We are not addressing that, we are assuming here that now; obviously, as you people what I will really like to see is that you are also being able to combine these two and come up with a new algorithm.

Now, this has got still a lot of scope of research, so here we are assuming that only one voltage can be supplied voltage is given and the overheads are negligible, overheads of what we are saying voltage switching is negligible. So, based on these assumptions let us see what the operating system will have to do.

(Refer Slide Time: 32:50)

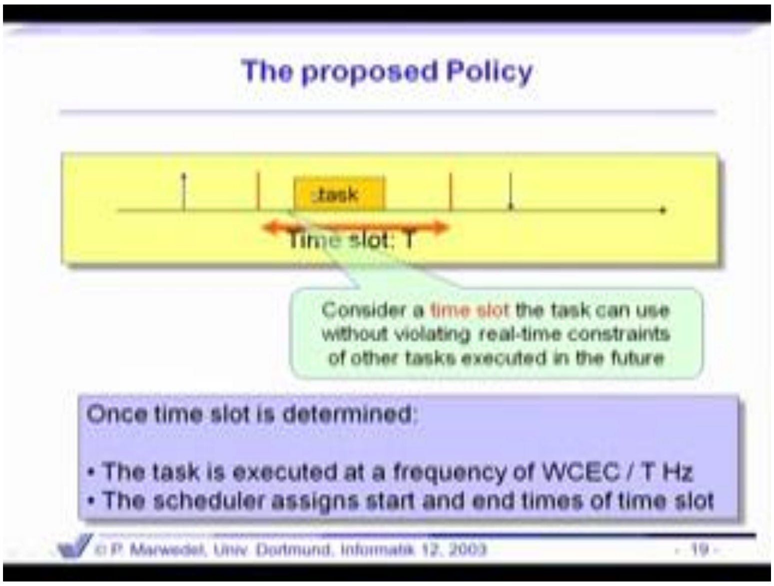


Look at this scenario, a task try to understand this representation these up arrows; arrows in blue are showing the arrival times, down arrows in blue are showing the deadline by which the task must complete. Now here typically we have shown that task 1 has been allocated to 2.5 volt which completing here and task 2, it has arrived here you see it came much earlier but we are talking of a single processor system. Therefore, I cannot allocate it I have to wait and only after this completes, I can start it. Suppose I am running it at 5 volt; if deadline was here it has peacefully completed within that time.

Now, task 3 arrived here where task 1 was still running and I could allocate task 3 only after task 2 has completed; I am right now not looking at priorities or any complications like preemptions and all those things. So, task 3 runs at 4 volt; I have made it to run at 4 volt; so that it is deadline is met, I could not run it at 2.5 volt in that case deadline would have; now if I had might be there was an opportunity I do not know that if this could be run at 4 volt, it would have extended and this could be run at 5 volt and could have met but that is how do I allocate it that has been found out by through some model.

So, the problem with the operating system is what is the optimum supplied voltage assignment for each task; in order to obtain minimum energy consumption for each task, it is not our example had only one task, only one task earlier this example where we the task of 10 to the power 9 cycles; that was one task and that one task also in that I managed the energy by switching it. Here I am dealing with multiple tasks; I have to decide on what would be the optimum supplied voltage assignment. Assignment means for which task? For which cycle? What would be the voltage?

(Refer Slide Time: 35:25)

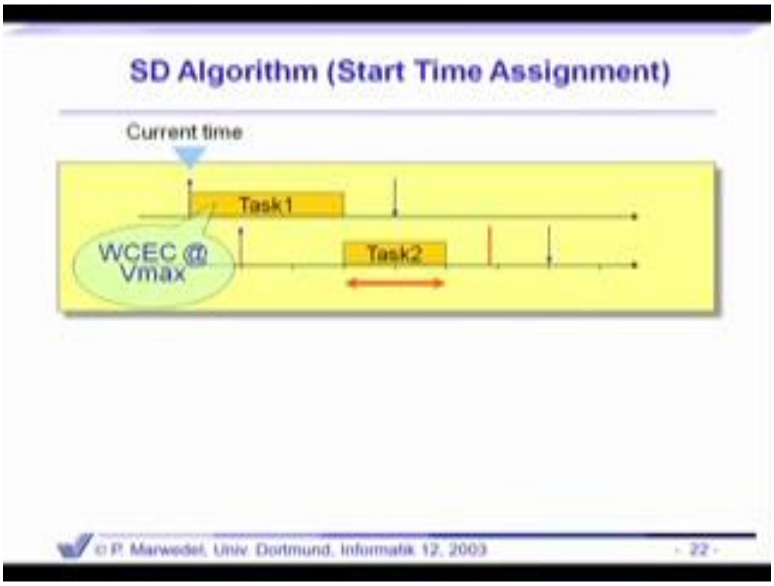


So, the proposed policy is this, so here is the arrival time, here is the deadline. So consider time slot, the task can use without violating the real time constraint. So, I assume this is the time slot within that if I can finish the task then it is real time constraint is not violated then once the time slot is determined, the task is executed at the frequency of worst case; now this time slot is T hertz; so much time, this is my time available within that I can manage and the deadline will not be missed.

So, now I try to run this task throughout this time slot, so it is worst case execution time was this. So, what will happen if I extend it to the supply volt? It will come down, so the scheduler will assign the start and end times of this slot. So, the task is elongated like this and I have reduced this is the over time, so the point is that if I can find the zone in which I can fit the task, then I occupy the entire zone by doing this.

So, scheduler then assigns the start and end times of the slot, the scheduler will tell that this is your start time, this is your end time and this start time and end time that the scheduler does till now in our normal operating system scenario, it was taken based on the complete throughput and deadline nuances but here we are interested in the power aspect as well.

(Refer Slide Time: 37:15)



So based on that there will be a couple of algorithms; I am not starting it in today's lecture, in the next lecture we will talk about a couple of algorithms which will be showing how the operating system can handle this.

Student: Sir the real time slot comes.