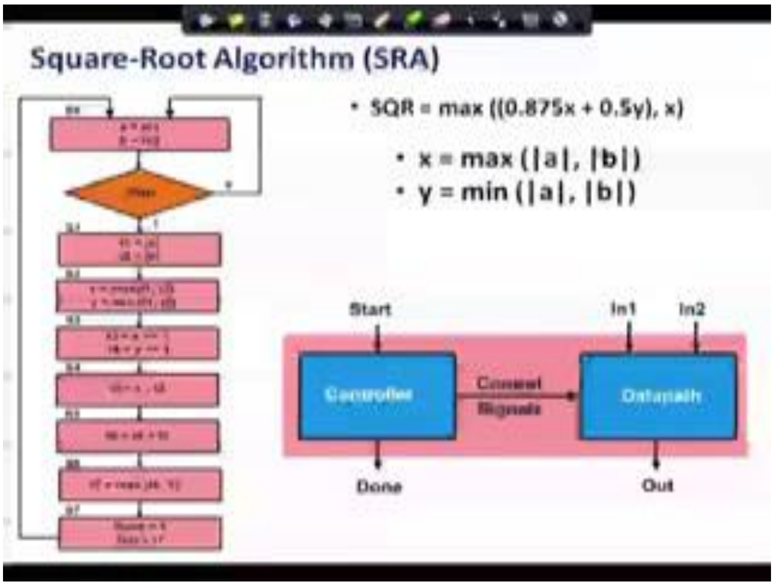
Embedded Systems Design
Prof. Anupam Basu
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

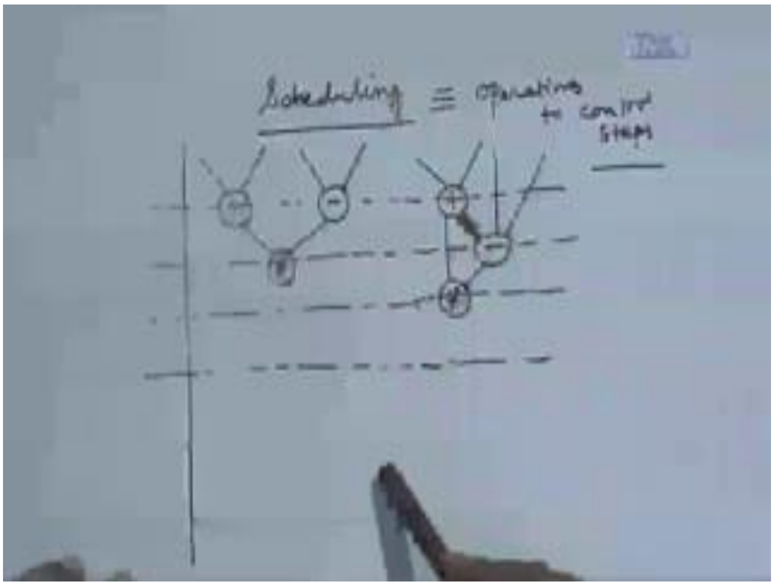
Lecture - 42
Scheduling

(Refer Slide Time: 00:23)



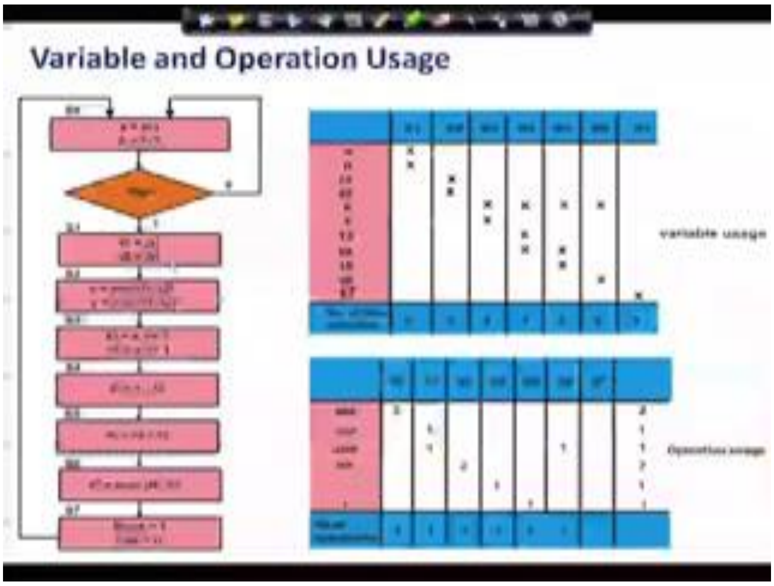
We have seen in the lectures till now that we can do some optimizations and reduce on the number of registers, number of operators, and number of interconnects; and at that point, we did not consider one very important thing that is known as scheduling or scheduling.

(Refer Slide Time: 00:43)



So, here you see we have boot up here say two operations in the same control step.

(Refer Slide Time: 00:57)

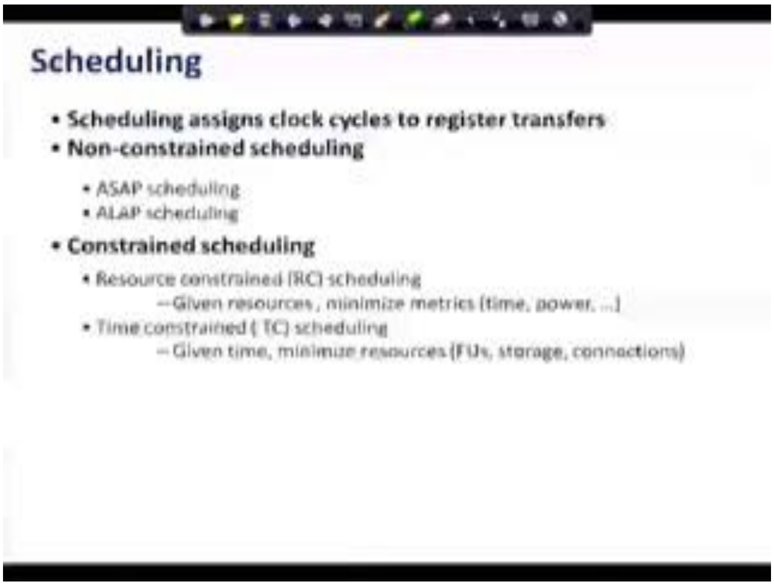


Now, before that let me just quickly explain what is meant by control step. Say, at every clock cycle some operations will be done. So, I can assume each of these to be the control steps. And say I can have an addition done here, and the subtraction done at the same control step. If I want to have them done at the same control step, then of course, I need a separate adder and a separate subtractor or two values; and maybe that then we come and do some multiplication. So, now I could have done this in a different way like if I want to save on the number of adder, subtractors, I could have done that I do the addition here and then do the subtraction here I am sorry the subtraction was not coming from here, so not this and the subtraction here I do. And then take the result of this adder and the subtractor, and do the multiplication here.

So obviously, if I delay this then this will also be delayed because of this dependency. At this dependency be not there, then probably I could have completed the thing earlier that I do this subtraction here and the multiplication in the same step. So, this is allocating the functional operations, the operations to the control steps that is known as scheduling. So, scheduling is basically a mapping of operations to control steps. So, in this case, as we can see two operations - two absolute computations max and min computations both have been done in the same control step, based on that we needed so many steps, so many states.

But today we will now look at how the scheduling can be done, so that different aspects can be looked into, one is the resources the number of adder subtractors that will be needing, and also the amount of time that will be required for a particular computation. Now, one thing you must understand that as was being shown here that the schedule that I do, we will also have an implication on the number of functional units that I will be needed. So, on one side, if I had say unlimited number of functional units then and if the dependencies were not there then I could have schedule all of them together. If the dependencies are there and limited number of functional units are there then my only constraint in scheduling are the dependencies, but in reality both of these are constrained I have to have some time there are dependencies as well as the number of functional units are not infinite.

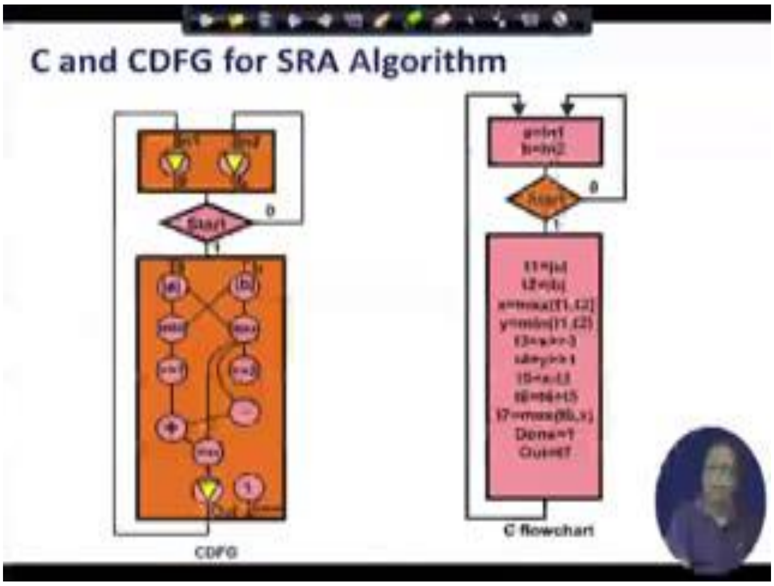
(Refer Slide Time: 04:57)



So, now let us move to the scheduling proper. So, as I said scheduling means assigning clock cycles to register transfers. Now, this is important, what I mean by register transfer. If we look at it as just in the last step of our earlier lecture, when we allocated the variables to the registers, we could see that all the operations could be represented in terms of data transfer across registers. We can have different types of scheduling; one is non-constraint scheduling, there is no constraint you just do some scheduling and there we have got two very popular ASAP and ALAP. ASAP means as soon as possible scheduling; and this one is ALAP is as late as possible scheduling.

We will also see some constrained versions of such scheduling; and we will particularly see two types of constraint versions one is resource constraint - RC where what we mean is given the number of resources, resources are known minimize the matrix, minimize the time and the power etcetera. On the other side, there is a time constraint scheduling, where given the time we might we minimize the resources. We have to satisfy a particular time; and respecting that, we have to try to minimize the number of functional units the storage connections etcetera. In the resource constraint, the number of resources are already told to you. So, based on that you have to reduce, you have to see how fast you can do that how we can minimize power etcetera.

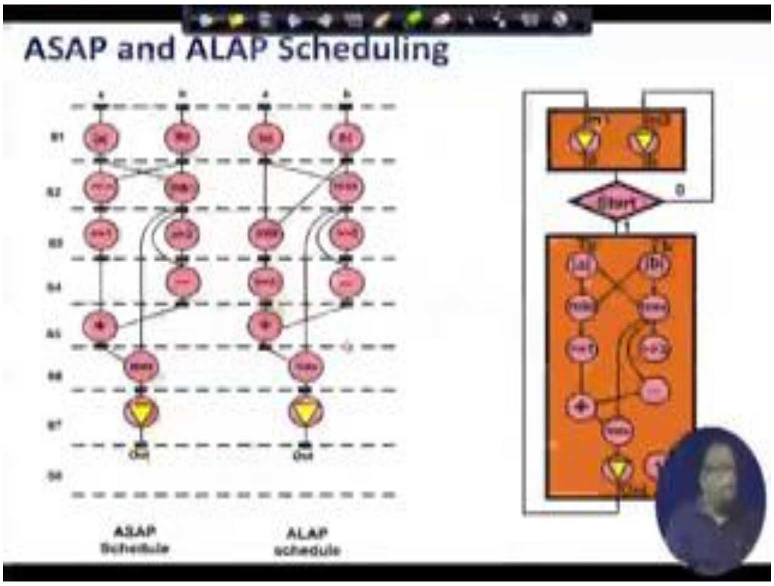
(Refer Slide Time: 06:55)



So, for here, we will look at a different representation here the same problem same square root approximation algorithm we are looking at taken from Daniel Gajski example. Here we are showing C flowcharts sort of C steps. And here we show a control and data flow graph - CDFG. Now, here in the CDFG, unlike the FSMD, where we have shown all the data operations explicitly, what we do here we only show the operators or the operations, we only show the operations.

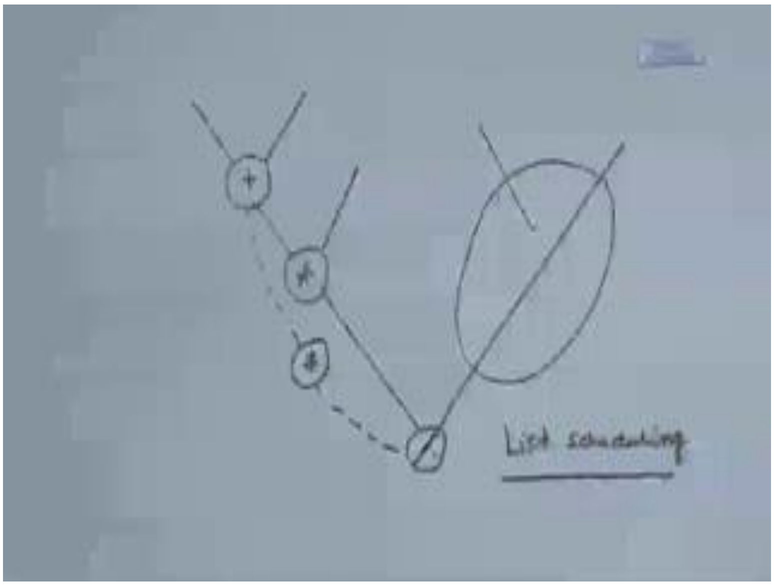
So, absolute finding of a and b, we are starting with a and b; a and b are coming as input here, min, max, shifting, addition, subtraction, max computation then we send one to done as a controller output and the output comes here and that is my CDFG. So, this entire thing can be transformed into a CDFG, where I just look at the data the variables are not shown here you see, the variables are not shown here. Here the variables are data that are flowing along these data edges, data flow edges, the data is flowing along with these operators.

(Refer Slide Time: 08:27)



Now, given this, here are two schedules. You look at this one this is our CDFG, this one. And what has been done in this case is the ASAP schedule as early as soon as possible, so what we did whenever what do you mean by as soon as possible, say whenever I have got the data for a particular operation ready.

(Refer Slide Time: 08:55)



Say for example, there is an operation add, and there is no further dependency on this. So, here the data is available I schedule it all right. And there is another operation which is actually dependent on this may be multiplication. Now, once this data is available, I can schedule it here or as since it is available, I can do it much later because the output of this is required maybe much later maybe with some divider something of that. It is coming to another path with some other graph it is coming here all right to other graph is here other sub graph is here. So, it is coming here. In my as soon as possible as soon as I get the data for this, I will schedule it. Although it is result is being used much later.

On the other hand, in as late as possible what I will do I will try to schedule it although the data is available I will schedule it as late as possible say immediately now everything is available. So, I have to do this, and I was delaying this further. So, I do the multiplication here, this is as late as possible, this as soon as possible.

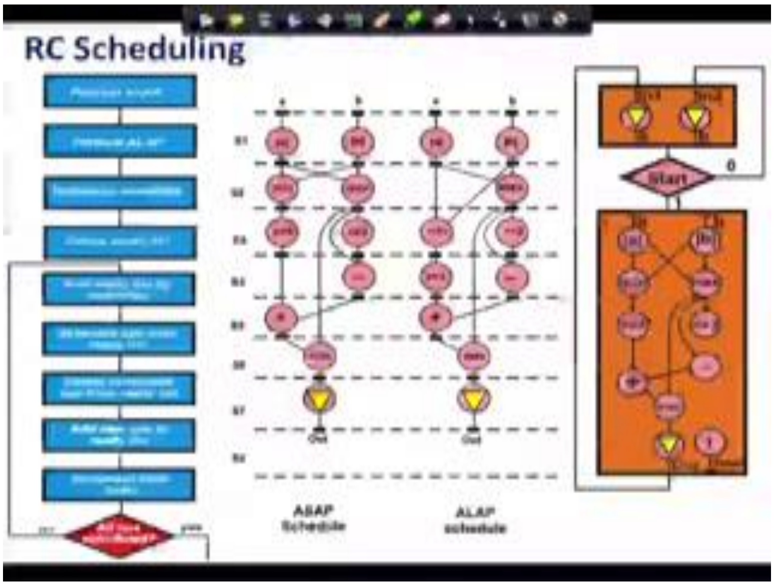
So, if we look at this how does it look like, how does it look like if I do this. You see I have got the a, and these are my control steps, all these dotted lines are my control steps. So, I am doing the absolute computation here both of them mean, now b is dependent on b and a. So, min can be done, max can also be done. Shifting of one is dependent on min, so I am scheduling it as soon as min done I am scheduling this here, but you see the output of this shifting is required by this plus which is rather delayed. So, the plus is actually cannot be scheduled before this, plus cannot be scheduled before this. Why? Because it is consuming data plus cannot be scheduled here I cannot scheduled here, because it is depending requiring the data from here. And this shifting will feed data to this, so I could very well move this here that would not have affected my performance, but still since I am doing as early as possible or as soon as possible I am scheduling it as soon as it is available.

On the other hand, here you see min. So, here let us look at this part in the as late as possible schedule, what we are doing is I need the max here. So, the plus will be done just before that as late as possible. The shifting is required in order to make plus possible, therefore I shift, this shift by one to this position by one, so it comes here. Consequently, min, it is responsibility is just to feed shifting, so min can also be shifted down right. And consequently, this is the as late as possible scheduling.

Now, if we look at this you can see that in this case since I have done as soon as possible if I have got some functional unit that can do min max at the same time. I will need two of them, whereas I can have only one of them here and that can be reused here that is possible to be reused, I can reuse this. Now, that is the pros and cons of the different scheduling algorithms.

Now, this as soon as possible and as late as possible is giving me another information. It is giving me the information about mobility of an operator. What is the mobility of an operator? There is a mobility of min, because min has got the mobility it can be either here or here. So, min has got mobility 1, whereas max has to be done there. So, max has got mobility 0, whereas I find that shift has got mobility 1. So, therefore, based on this ASAP and ALAP, I can also get an idea of mobility. So, it is serving me two purposes either I can do an ASAP scheduling as early as possible I finish it off; and then pay the cost for the resource user. Or I do it as late as possible and see if I can save some resources not that in all cases that will be beneficial, but sometimes it will be beneficial. The other advantage that it gives us is it shows us what is the mobility that I have got for the different operators.

(Refer Slide Time: 15:02)



Now, given this, we can come to now resource constraint scheduling. Suppose, now I have got some resources constraint, I mean resource constraints already given to me. So, for that this algorithm is also known as the list scheduling algorithm it is also known as in the literature as list scheduling algorithm. What is being done here first time performing the SR, and performing the LR, and then I am finding out the mobilities.

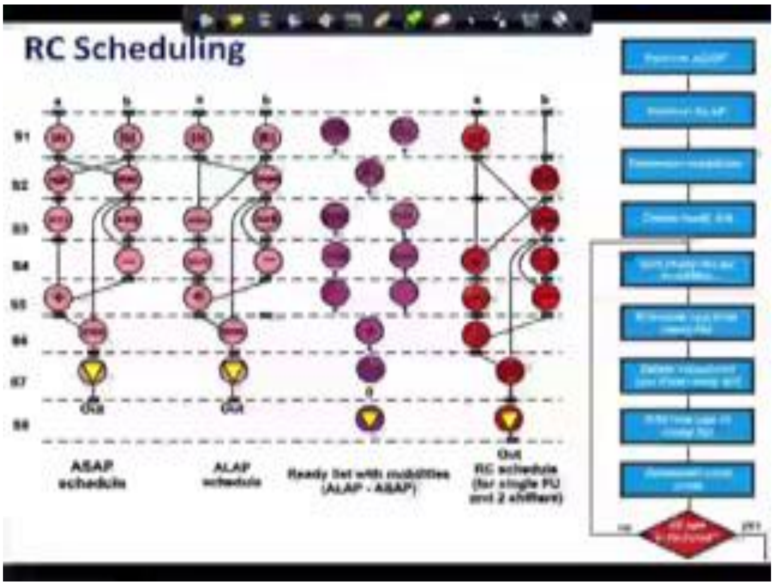
So, I do the same thing that was shown in the earlier slide that I am finding the mobilities of max, min everything. And then create a ready list ready list means which are the functional elements which are ready now that means, all its data are available. Then from the ready list we will sort the ready list by mobilities, with a least value of mobility coming earlier and the highest value of mobility going later, so that is how I will make the list in the list scheduling algorithm.

So, I make the list scheduling list will be based on the mobilities. So, I sort the ready list by mobilities then I will schedule the operations from the ready list. Ready list means, I mean the list that I have made. Once I scheduled them I delete them and then add new operators, once I scheduled them they are scheduled, so that is the ready list. What I mean is so for example, here right now; what will be in my ready list initially can you tell me, what is ready at this point? A and b are ready min and max are not ready as here therefore, a and b will be in the ready list, and they will be sorted by mobility. What is the mobility of a and b, both are 0, therefore, I can take any one of them.

Now, as soon as a is scheduled I assume that a has been computed. What do I add to ready list I do not add anything to the ready list because of the dependency. Next I schedule b, and once b is scheduled then what is what comes in the ready list min and max both coming the ready list. Now, out of this min and max which one has got more mobility, min has got more mobility, max has got less mobility, therefore, max appears at the top of the list and min below. Therefore, I now schedule max.

As soon as max is scheduled, as soon as max is scheduled what becomes ready what is there in my ready list, min is already there in my ready list with mobility 1 and now shift comes to my ready list. Now, shift has got lesser mobility, therefore this shift will be scheduled and then this shift makes this minus, but then I have got my ready list. So, I will schedule min also, because as soon as I come here min becomes again the no further mobility. Therefore, min will be scheduled. In that way we go on in the list scheduling algorithm that is how the scheduling is done. And here given my constraints and every time I see the whether all operands are scheduled all operations and op codes are scheduled.

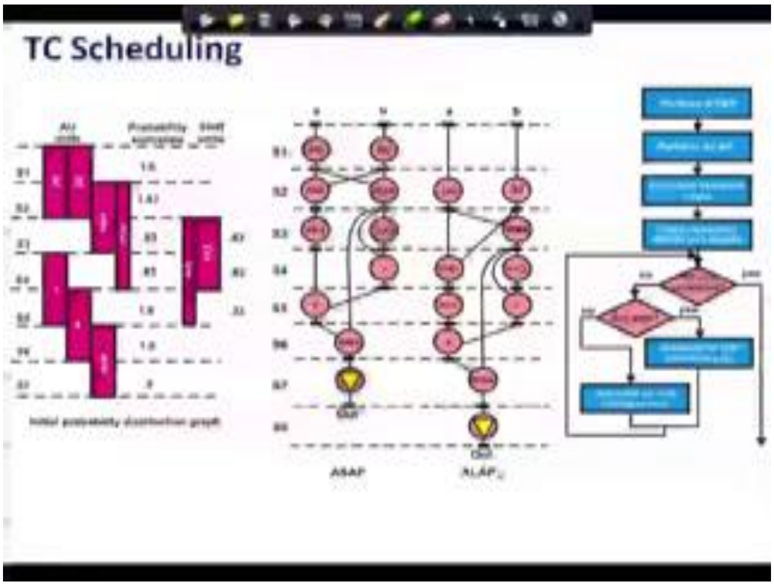
(Refer Slide Time: 19:02)



So, here you see the same thing ASAP and ALAP the thing that I was explaining to you right now is that a and b were ready, a and b were ready, but I schedule to a. Now I am doing a resource constraint schedule where I have got a single functional unit, a single functional unit and two shifters. And the single functional in unit is doing all the things plus, minus, max, min absolute computations that is my resource constraint given that resource constraint my schedule will vary. If I had two functional I mean two multi functional units then I could have my schedule would have differed. So, here it is one and b is ready b is scheduled here.

Now, you see I am showing the mobilities here min has got a mobility 1, so after that max is scheduled here. And min therefore, comes here and three is scheduled. And now I have to do min because otherwise they cannot schedule this shift, and in that way I get that. So, here is the ready list ASAP minus ALAP, and here is the output RC scheduled that we get. This is the resource constraint schedule that we get as a derivative of ASAP and ALAP, and given some resource constraints this is the list scheduling algorithm which gives us this schedule. And here we find that we need eight steps and we are getting the output at a eight step.

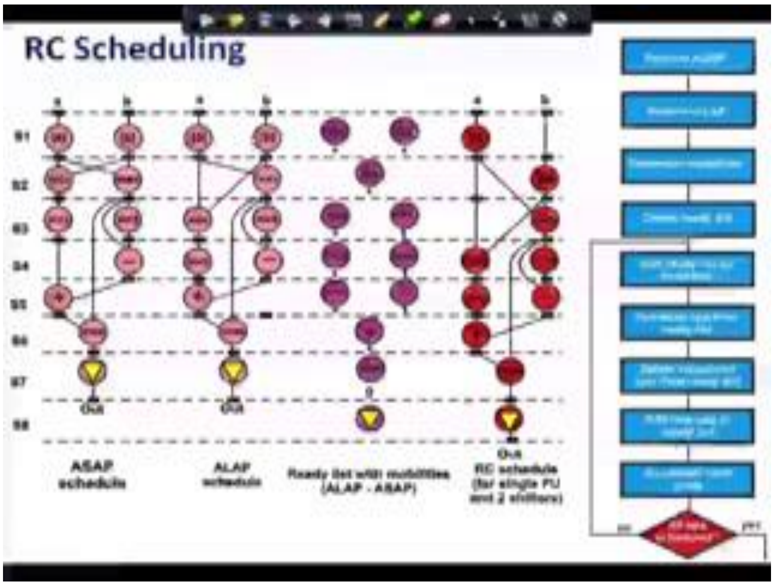
(Refer Slide Time: 21:18)



Now, the constraint could have come in a different way also the constraint could have come that I am not giving you any resource constraint, I am giving you time constraint that come what may whatever resources you need to take it. But I want the output by within eight steps eight control steps. For that the algorithm is again let us first look at the algorithm, we perform the ASAP and ALAP, and then determine the mobilities. Now, we are taking recourse to another approach which is we create the probability distribution graphs.

And based on that we go on scheduling the operators, if all our operations are scheduled fine, otherwise we will go for the maximum gain, where do you gain maximally. If they are same then we go for the minimum loss, and we carry out this algorithm. We will show it through an example the same example that square root approximation algorithm is coming in very handy with us. So, here you seen a and b. Now, I have got my ASAP here this is my ASAP schedule right, this is my ALAP schedule. Now, this ALAP schedule is different from the ALAP schedule that was shown earlier.

(Refer Slide Time: 22:40)



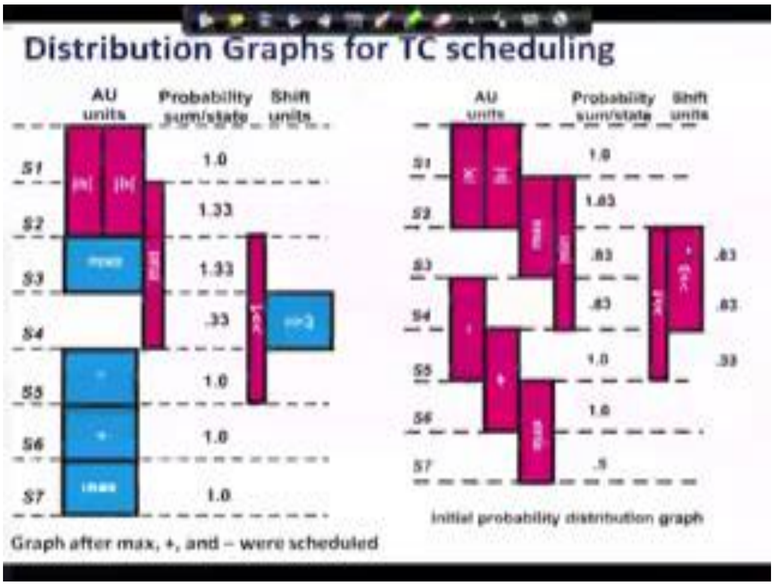
This ALAP schedule is different, because you see this ALAP schedule completed the operations at control step 7, but here my time constraint that has been given is control step 8, I am more relaxed at the time. Therefore, if I start my ALAP schedule, I will take advantage of that. I will start with 8 and then schedule max in 7 here I started as early as possible and I finished to this 7. Here I start with 8, and go backward therefore, I am scheduling the max here, because max is the only predecessor for this then plus can come here because that is fitting in this, everything is shifting down ultimately I can delay in this a, b.

Now, given these two now, if I will look at a sort of probability that where can probably a b scheduled sorry where can probably absolute a b scheduled the absolute operation on a, absolute operation of a can be scheduled at either s 1 or at s 2, so both of them are possible. What about b, b can also be scheduled either here or here. So, each of them have got are contributing to 50 percent probability. So, at this step s 1, the probability sum is 1, because either this will be scheduled or this will be scheduled. This maybe scheduled or this maybe scheduled. What about this, max can be scheduled either here in step 2 as shown here or max can be scheduled in step 3.

And min can be scheduled in here, here or here, clear follow this. Min can be scheduled according to ASAP here; min can be scheduled according to ALAP either here or here. So, min has got this range. So, min is contributing to about 0.33 probably to all these steps therefore, at this step s 2 my cumulative probability is 1.8, it is not no longer a probability in that sense it is 1.83 is the value. Here max and that is saying that something what is the probability that something will be scheduled here, so you can see that here 0.83 because max can be scheduled either here or here min can be scheduled either here or here or here. So, max is contributing to 0.5, min is contributing to 0.33 to this control step.

Similarly, for minus this should be minus I am sorry it is, it will become vertical minus plus and max are like this. So, once we have this then we can see that we have to scheduled something I have to come out up to this state, and I have got my resource constraint here there is a time constraint no longer resource constraint.

(Refer Slide Time: 26:35)



So, how can I proceed with this with this now I decide that if this was my initial probability distribution graph. Now I have not discussed about the shifting. Shifts you see the shift could be done either here at s 3, s 4 or s 5. So, it is chance of being scheduled is an either of this shifting by three is in either of these two. So, you get the values. Is this clear? Now given this.

Student: (Refer Time: 27:12).

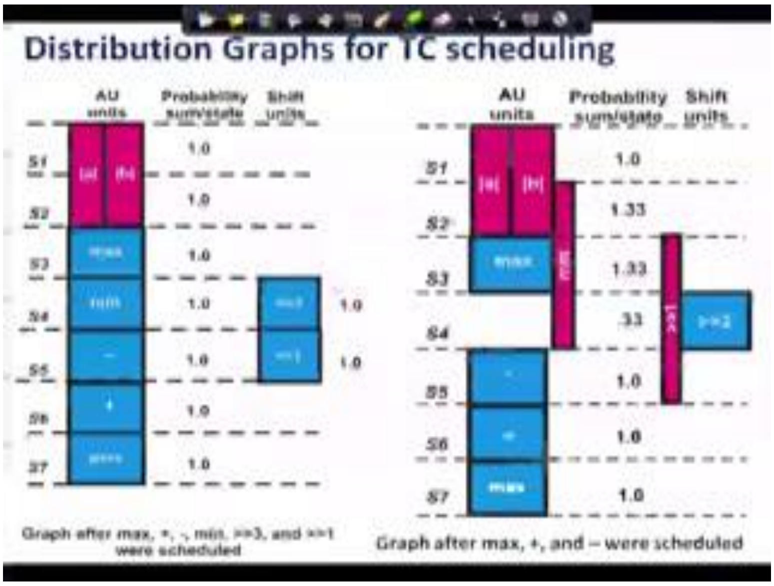
Pardon.

Student: (Refer Time: 27:13)

Professor: This one, this one, so you see shifting by one can be either in s 3, s 4 or s 5, so that is 0.33. And this one shifting by three can come either here or here, so 0.5. So, over here 0.5 and 0.33 only for the shifter, it is 0.83. So, now given this now I have got the max here now max certainly I can schedule here, because I am time constraint and meeting my constraint, meeting my constraint. I can also think of that if I put max here then I am also not conflicting with any other resources here, because although no resource constraints have been told to me, but for economic design, I should also look at the resource minimization. So, I can also combine these parameters. So, max is being scheduled here now. So, max as soon as max is scheduled here, certainly max is schedule here, so that is one max is scheduled this step is now full. This blue means wherever I have already allocated so that means, max is full.

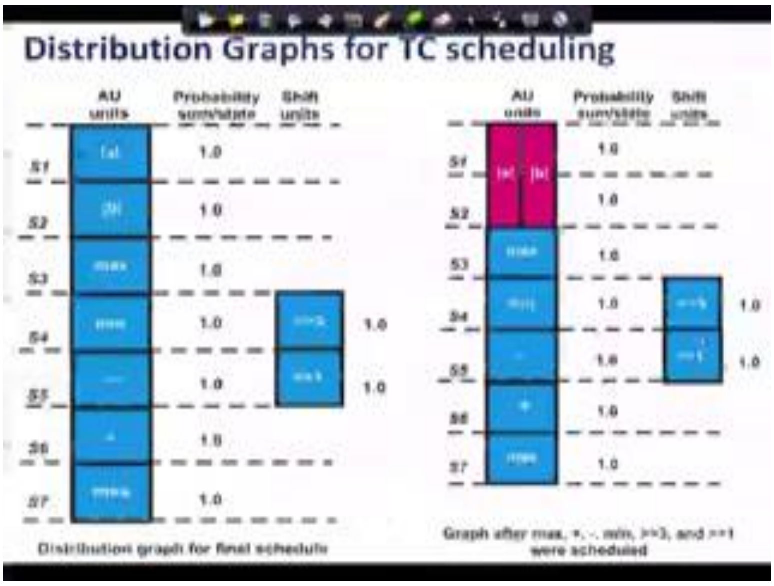
Now, next I will do that plus can there be here or here, now this time is free. So, I put plus here. So, this is also full minus here, it is also full. Now, as I go to this minus, so these are full, all full. Now, this part has got a probability of 0.33, so I can put min here or I could have put min here, I could have put max here. So, out of these now I have to schedule. Now, again from here I can see that if I put shift by 3 at this step, because of the dependency you can see that if I look at the control flow graph this shift is required for this subtraction. Therefore, I have to schedule shift before the subtraction as soon as this has been done this will actually apply a force to this that you now come. So, therefore, this has to be scheduled at this step once this has scheduled at this step, now next I scheduled max here min is yet to be scheduled and a and b are still left.

(Refer Slide Time: 29:53)



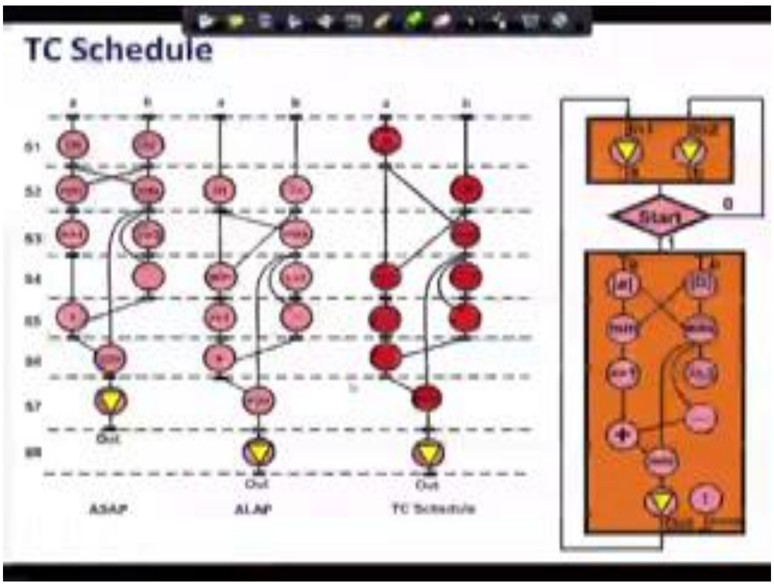
So, next I move. Now, here max was scheduled in s 3. So, s 4, I can schedule in min. So, that everything becomes you see 1; a and b now so this was after max plus minus was scheduled I got this. So, max is put in s 3 min comes in s 4 min comes in s 4; and a and b now what is the situation here, it was 1.33 here. As soon as min is scheduled here both of these become 1, now I can schedule them in any one of these. So, and this shifting I can scheduled anywhere. So, I do this a, b and this one, this one, can now instead of loading this I can since that is let us look at the control and data flow graph once again. And on what does this shift to I mean the shift by one is affecting whom, shift by one is affecting this plus, with this plus which has been scheduled in s 6. Therefore, I can schedule it anywhere here. I prefer not to schedule it here, because in that case I will have to share the shifter.

(Refer Slide Time: 31:16)



So, I shift it here, here there is a shift operation and a minus and this plus will need this. And now only a and b are left. So, I can simply scheduled that in any particular way like a and b, so that is the ultimate distribution graph that I get for the time constraints schedule. Is it clear? So, again summarizing what did I do, I first had the control data flow graph and based on that I did the ASAP and ALAP found out the mobilities. And then after finding out the mobilities, I computed the probabilities of allocation; and based on that, I try to balance the priorities and ultimately everything should be one. And of course, here there are two operations.

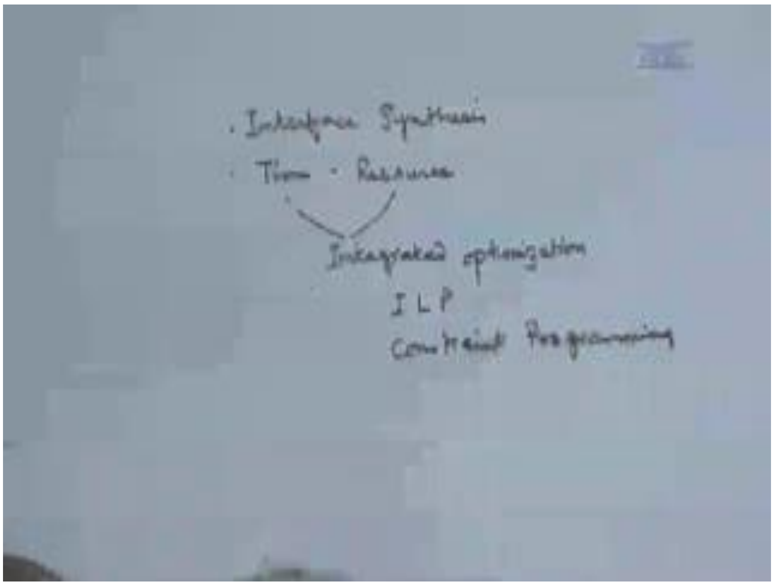
(Refer Slide Time: 32:15)



So, ultimately I get this schedule you see this is the time constraint schedule that I get. And now can you tell me how many function units I am needing? Now y 1, I will need a shifter, here these two are overlapping. So, I will need a shifter and one functional unit, one more functional unit. So, you see I had done the resource constraint scheduling earlier here with single FU and two shifters. And there I had since it was resource constraint, I went up to eight steps, but for time constraint also I have come up to eight steps and use minimized on the resources. But here there is another advantage you can see that this shift by 3 and shift by 1, if I agree to pay the cost of a general shifter, then I could have reused the shifter. I could have reused the shifter and thereby now it is a question of economics whether I save money or not on this.

So, based on that, we conclude our discussion on scheduling. So, we have looked at ASAP scheduling, ALAP scheduling, resource constraint scheduling and time constraint scheduling, so that is a brief overview of the high level synthesis of the hardware synthesis part the approach that is taken. I have not mentioned about many things which are also important.

(Refer Slide Time: 33:59)



I am just trying to note some of them here; very important thing is the interface synthesis. In the same way, you can see how the channels and the ports are allocated. Interface synthesis is there. I have not done see this is a classic optimization problem where the time and resource conflicts are always there. And if I so integrated optimization, optimization is a very important issue; integrated optimization for that people take recourse to integer linear programming, constraint programming and other approaches. So, based on that there are number of optimization works that you can find in the literature, but this is just to provide you an overview of the nuances that are involved in hardware synthesis.