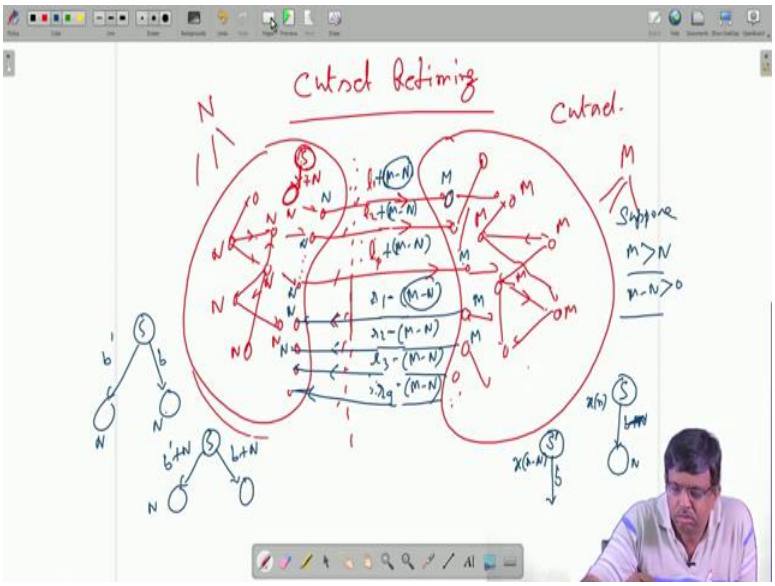
VLSI Signal Processing
Professor Mrityunjoy Chakraborty
Department of Electronics and Electrical Communication Engineering
Indian Institute of Technology Kharagpur
Lecture 08
Cutset Retiming

(Refer Slide Time: 00:25)



Okay, so this is also our figure and now what we do is this, we have, for doing retiming, we have to assign some retiming values to various nodes, we cannot violate that that is the Retiming Theory. And we have to make sure of the feasibility, that is after retiming, no delay of the node, after retiming should become negative. That we cannot violate.

We have to follow Retiming Theory that is we have to assign retiming values and based on that for retiming formula only, we have to change the delays and make sure that nobody turns out to be negative, no delay amount turns out to be negative. But while doing so, maybe I'll, in the case of cutset retiming we proceed in a particular way. You choose one integer, may be capital M for all the nodes here, all the nodes here and another integer N for all the nodes here, okay. That is this gets N, this gets N, this gets N, this gets N, this gets N, this gets N, this N, not the source node.

Remember in the Retiming Theory we say that source node has a universal retiming value 0 that is it remains 0. The delay here, whatever be the delay in this edge, that gets added with the retiming value of this node; source then this, remember in the Retiming theorem we had a

edge from the signal source to a particular node and that edge had a delay bi, after retiming it became bi plus, retiming value of this node.

So, here whatever be the delay, with that you have plus retiming value of this node which is N. So delay here in this edge will go up by N. But how about the delay in this, say between this node and this node? That will remain unchanged because it will be original delay in this edge plus capital N minus capital M. So internally these edges between various nodes in one sector, this sector will not undergo any change. They will remain as it is.

So, I do not have to change the delays inside this sector between every pair of nodes except when it comes to source node, source node going to any internal node, if we that any amount of delay or zero delay I have to add a capital N to that, I have to add a capital N to that, alright. Because as I told you there during Retiming theory we have shown that if there is a signal source that is giving a signal to any node and that has a delay bi, after retiming bi will become bi plus, that is plus, whatever be the retiming value of this node. In this case it is N.

So, only this will go up by N, but all other edges between every pair of internal nodes, they remain same. Here again the same thing, everybody gets M, M, M, M is your choice, M, M and like that. So, since both destination and source nodes for this edge have got retiming value M, so original plus M minus M so it remains original.

So, no change in all the internal edges, in the delay in the internal edges. So, and there is no source node I have taken to this side, a signal source, whether pushed to this side, it is perfectly unchanged inside this sector. Changes will come in these edges. Let me remove this D3, we will know D means delay. So, now you seek versus this edge.

What will happen to this edge? Original yellow one plus this is the destination, this has M, this is source, this has N. So, it will be l 1 plus M minus N, same for these, l2 plus this has M, this has N, everybody here has M, everybody here has N. So, factor will be M minus N. This also, this is going to M, N so M minus N by this, this will be added to this.

And here what will happen? This is again N, this is M. But this is the destination, this is the source. So, it will be r1 plus N minus M, N minus M or r1 minus, the same figure M minus N. Similarly r2 minus, either r2 plus N minus M or you can write same here. So, suppose M is greater than N, your choice, so M minus, suppose then N minus M positive.

So, to every edge on this, on this (direc) in this direction you are adding a positive number of delay and to every edge in this direction, from every edge in this direction you are subtracting the same M minus N, same positive number. We will just make sure that nobody after this subtraction here in this direction becomes negative. That is the only thing. Then there will be no feasibility.

But choice of M and N is yours. So, in one direction you add some extra delay, if all the edges in one direction and take out the same amount of delay from all the edges in the opposite direction. Within each sector do not change anything. Now, comes to this N. Now, here let me explain that part in a separate way, in a separately.

Suppose you have your S. You had original delay, may be you know I mean some b and this is a node. This has retiming value N. So, I say b plus N, and it was generating x of n. So, what is arriving here is x of n minus b minus capital N. What you can do? You can modify the signal source to be S prime, which generates the delayed version of the signal. So, out of b plus N, this capital N delay already absorbed the S prime, so this delay remains b. So here again b doesn't change.

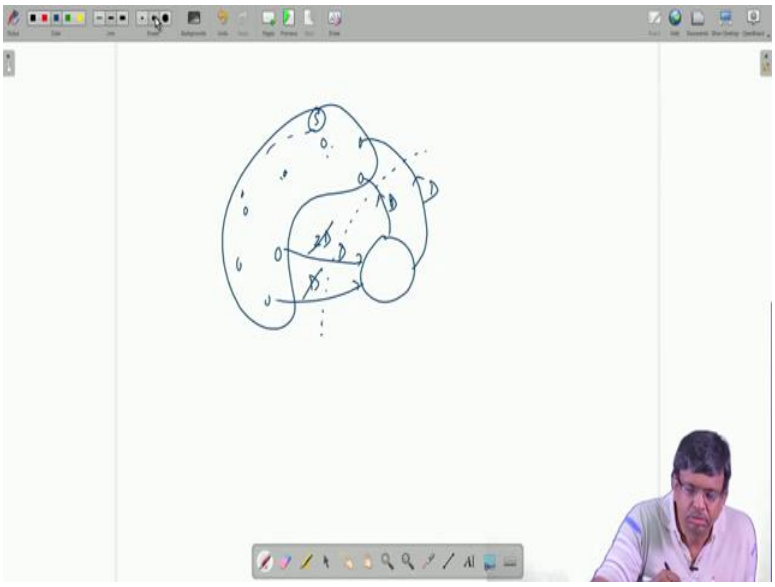
Signal source same set, I mean produces a delayed input and since there is a time invariance circuit, all node outputs definitely also will be delayed by same amount, but which is fine with me because you know, N is known to me, alright. But in this case if it is b originally this also remains b after retiming.

If you have got things like you know one source but it is giving it is signal to two different nodes, this has retiming value N, this has N, but this has b, this has b prime. So, now after retiming this will become b prime plus N, because this is N; and this will become b plus N, alright. Again there is no problem because both are N, so both, in both cases the delay has gone up by the same factor N so you can absorb that like this, signal source will be delayed source and then you know, b will remain b, b prime will remain b prime.

So, inside actually there is no, you do not have to change anything. Only change you have to make is is the connecting edges from left side to right side and right side to left side. In what direction you add the constant figure say minus N, in opposite, from opposite direction you subtract. So, you get a new DFG with new delays into these edges. Then again you look at the DFG.

Again by visual inspection, identify some new cutset, that is new edges which from another cutset. Again apply the same there and go on doing it but if your target is to minimize the original critical path, you have to be clever. You have to target the critical path and choose your cutsets, cutset edges such that eventually it leads to minimization of critical path. That is essence of Cutset Retiming. Bigger problems here can be tricky, problems can that be tricky so you require some practice. Maybe I can take few examples now.

(Refer Slide Time: 08:56)



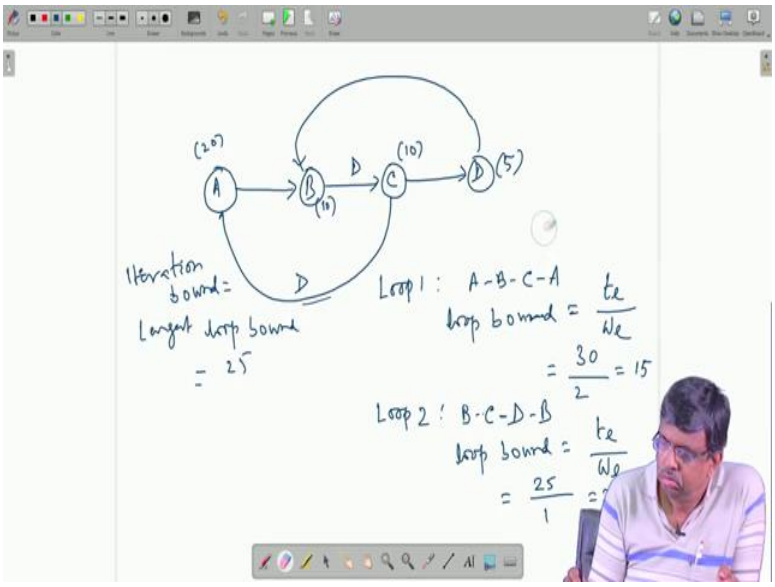
Consider something like this, oh, just a minute. Suppose, there are things like this and a single node you have taken out, this is a single node and this has got various nodes, various nodes. Signal source also maybe this side, no problem but one particular node I have taken out, so from this side there are some edges; in this direction also there are some edges.

In one direction, in another direction. May be you have got 2D and D and you have got nothing here. See my dotted line now, dotted line is cutting this edge, this edge, this edge, this edge. Clearly they form a cutset because in your mind you imagine that all the 4 edges had gone. You have removed them. You have cut. Then you will be left with one node, single node which is also a valid sub DFG.

A single node is a minimum DFG and of course another sector which is internally connected, I mean, various nodes they are internally connected, that is another sub DFG. So, just two, original is cut into two and therefore these four qualify to form a cutset. Now, this is one direction. I have got 2 D and D, this is another direction, I have got nothing. So, I can take up may be a constant factor one delay from here.

So, D will become 0, this will become D constant and there is one D I deduct from here, one D, same one, I deduct from this edge also, this edge and same one I add to this side. So, it becomes D and D. These are very elementary cutset retiming example, alright.

(Refer Slide Time: 11:26)

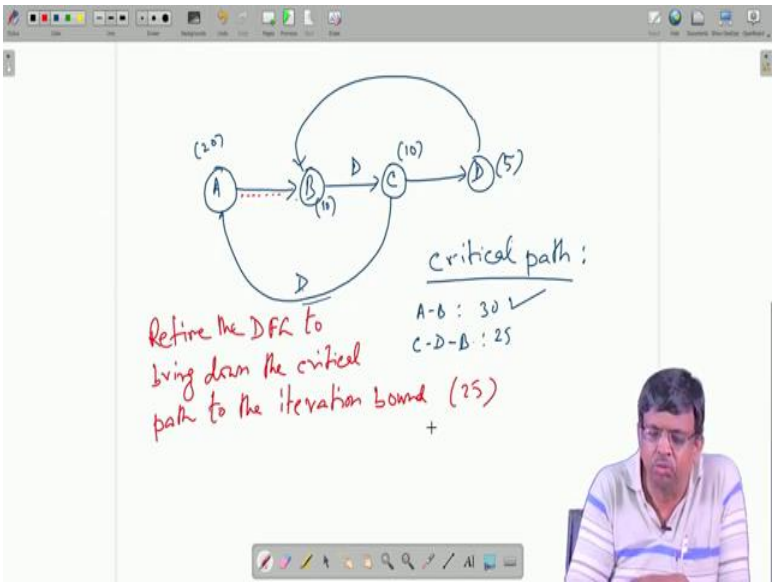


Take another example. It is from (())(11:22) book only, a beautiful example. Suppose I have got a DFG like this. This is node A, this is node B, this is node C, then node D. A takes 20 amount of time, 20 whatever be the unit, microsecond or pico second, nanosecond, whatever, this takes 10, this takes 10, this takes 5, alright. This has a delay D, and this has a delay D.

First question, what is the iteration bound, iteration bound? For iteration bound you first identify the loops. For each loop you find out loop bound. So, there are two loops here, loop 1 is A to B, B to C and C to A. So, A B C A these are loop. In this loop, loop bound, in the previous half I have done it, loop bound is t l; that is a total loop computation time by total loop delay. t l is how much? This is the loop.

So, 20 plus 10 plus 10, 30 and how many delays in the loop? 0 plus 1 plus 1, 2. So, this is 15. And there is another loop, loop 2, B to C, C to D, D to B. Here loop bound tl by wl and tl here is 10 plus 10 plus 5, 25, divided by how many delays, 1 plus 0 plus 0, 1, so 25. So, for iteration bound, you have to take the highest, largest loop bound; that is your iteration bound. So, that, between 15 and 25 it is 25, so this is iteration bound is 25. Let me erase this now.

(Refer Slide Time: 15:03)



There is critical path. Critical path, critical path means you have to find all the delay-free paths and find the total computation time along the edges, remember the nodes that come in that path, and then choose the one which is highest. So, let us start with A. A takes 20. Take the edge close to B this has no delay.

So, the time of A and time of B will add, this is part of the critical path, 20. Then if I move forward, there is a delay so I cannot move forward. One choice is A to B that is 30. Then you start with B. Of course you cannot go to C so B is 10. So, between 30 and 10 it is included in A B, so forget that. Then start at C, C there is one direction here; but this has a delay D.

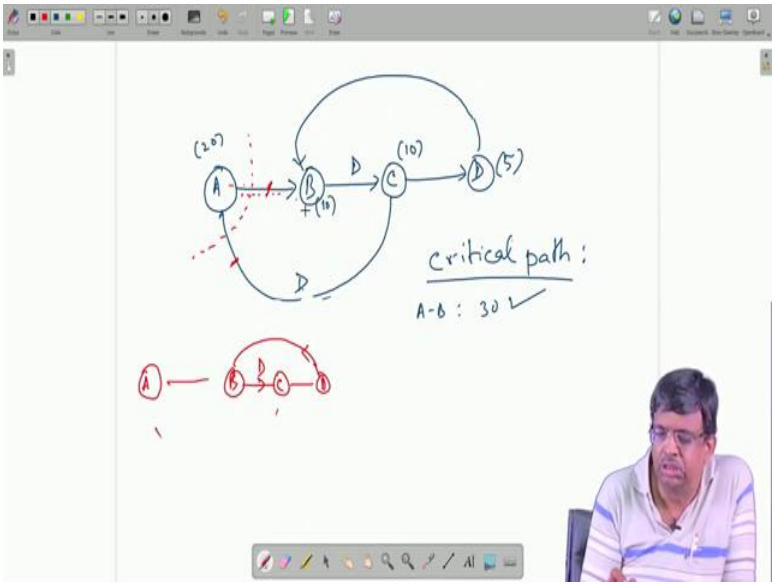
But there is another direction. So, if you proceed in this direction you have to leave at C only so time is 10. But if you want to go in this direction C to D, no delay. So, the two times will get added, 10 plus 5, 15. From D to B, another 10, so 25. 10 plus 5 plus 10, after that you cannot proceed because of the delay. So, this is another delay-free path. So it is 25.

So between them this is the largest, you have to take this. This is my critical path. I show by a dotted line, a critical path. Here the question is, you retime this circuit, retime this DFG so that critical path can be brought down to the lowest possible value that is the iteration bound. What was the iteration bound we found? 25. We found iteration bound to be 25.

So question is, retime the DFG to bring down the critical path to the iteration bound which was 25. Why iteration bound? Because we have understood, we have seen the meaning of

iteration bound that is the best you can achieve by retiming. You cannot bring down critical path of the loops below that. So, iteration bound 25, current critical path 30, question is try to retime it so that after retiming, critical path comes down to 25 because 25 was the iteration bound we just calculated a little while ago.

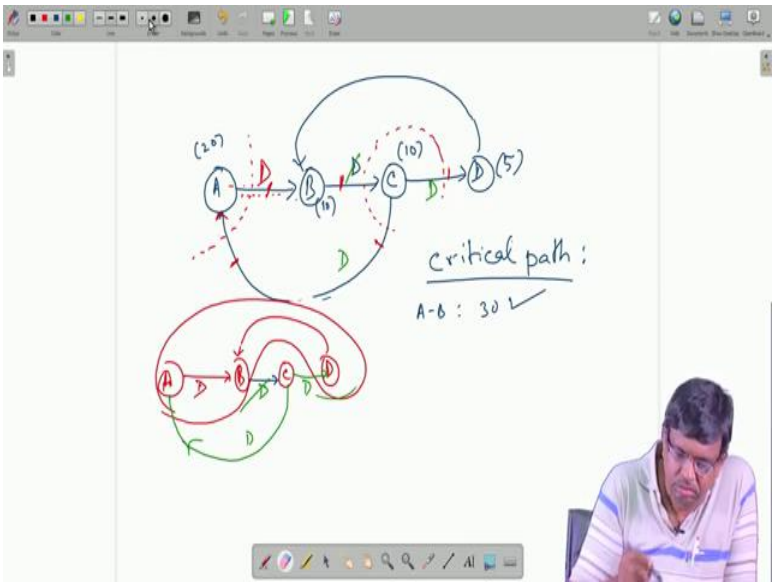
(Refer Slide Time: 18:33)



So, let us see how to do that by using cutset, and we will follow cutset retiming technique. So, what is the critical path here? A to B, 20 and 10, 30. This is the culprit. So, I have to have a delay here, because otherwise these two will get added as it is now, it will remain 30. I cannot afford that because I have to bring down the critical path to 25 and this is already 30. So that means I must have a delay here to separate the two nodes. So, how to do that? If I do this, see the dotted line is cutting this, this one and this one.

Do they qualify to be a cutset? Answer is yes. Because if you remove this A to B and if you remove C to A, then on one side you are left with A, it is like this, you are left with A, and you are left with another sector, if these two. So, this is one sub DFG, this is one sub DFG, which means these two qualify to be a cutset. But this is one direction from this sector, from this sector this is one direction and this is another direction, from this sector to this sector, opposite directions.

(Refer Slide Time: 19:47)



Therefore, from one direction I can take out a constant delay and add in the opposite direction. This edge is one direction. This has a constant delay. Suppose I bring it down to 0. So, then I can have one delay added here, D. This is the delay I was looking for, I wanted to have a delay here. So, that the two nodes are separated in time, the two, 20 and 10 do not get added. So, I may feel very happy.

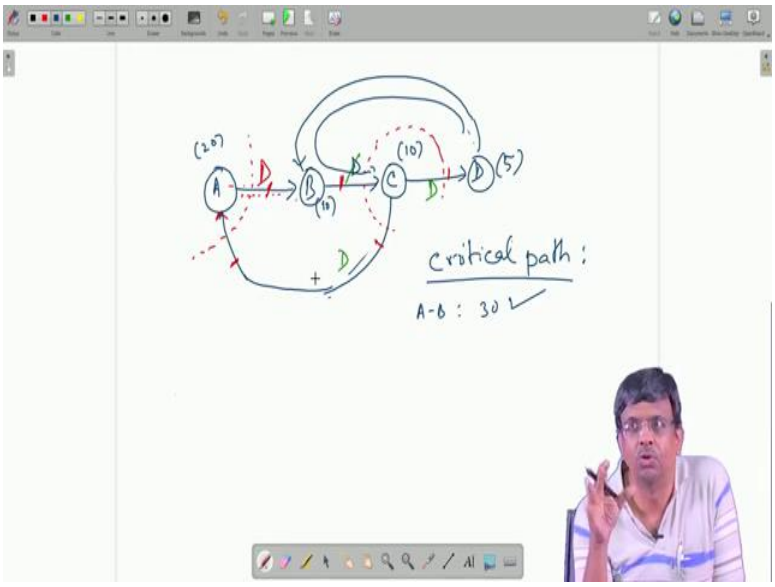
But we see what is happening here now, earlier this edge had a delay. So the C and A, their computation times were not getting added. They were not in the same cycle. First you carry out some job in C, then output is processed here in next cycle. But now this is a delay-free edge, the two times have to be get, have to get added. So, now again 10 plus 20, so 30. So, earlier 20 plus 10, 30. This was a critical path. Now, this is the critical path because delay has moved out.

So by removing the delay from here to here, I have taken care of this edge. It is no longer 20 plus 10, 30. But this has opened up. C to A has opened up. Now, the two times are getting added. It is becoming 30. So, is it any good? Shall we still stick to this? I will say, "Yes, I will still stick to this."

But then how to take care of it? Then comes my next step. I consider this dotted line. So, this is cutting this, cutting this, cutting this. Now, in your mind again you use imagination. You are left with one node separately and another part like A, B and D; A, B and D. There so delay, no problem and C another node.

So, this is one sector and this is another sector. So, you have got this delay, this is one direction from the bigger sector and you have got from C going to A and from C going to D, these are in opposite direction. From C to D, this bigger sector and this is from B to C. Now, B to C has one delay. And in opposite direction there is no delay. So, I can take out one delay from here and bring it here, add in this directions. That means this delay goes, it again comes here, it comes here.

(Refer Slide Time: 23:20)

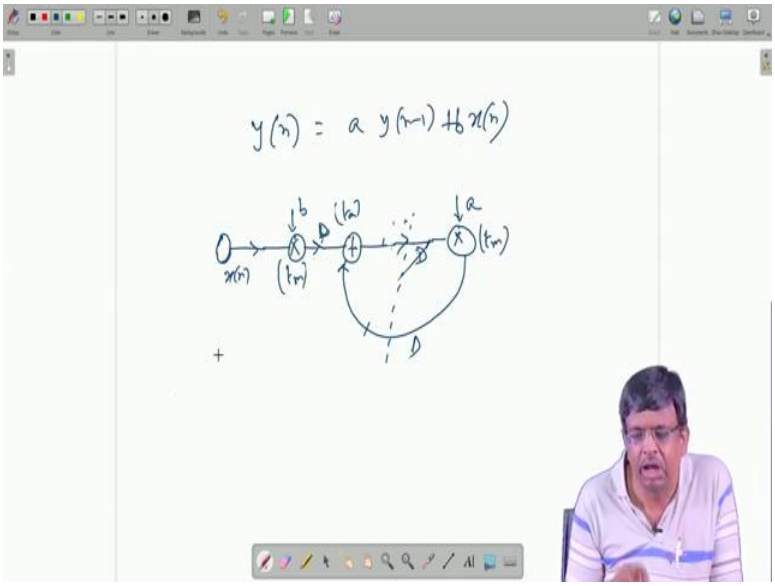


So now... So, now you see what is a critical path? Earlier I had a problem, this edge opened up. There was a delay earlier, now the delay gone but now again that delay has come back. So, this 10 and 20, they are not getting added. This 10 and, this 20 and 10 not getting added. And there is delay here. So, now what is the critical path? Critical path will be this one.

You started D, no delay in this edge, then B, then no delay in the edge, C. So, 5 plus 10 plus 10, 25 that is what we wanted. This is retiming. There are several examples in various textbooks including (())(24:05) book. You can try some. I will take up some example of what is called adaptive filters, you know, that because there is a key component. And there are some complications and I will take it up the issue of retiming that.

That will give you lot of insights as to how to carry out retiming in practical circuits and that requires some time. I have to introduce you also to the topic of adaptive filter little bit. But take another example. I don't know whether I have time to complete it today but we can just start.

(Refer Slide Time: 24:40)



Question is on IIR filter. y n is equal to a y n minus 1 plus x n. Very simple, maybe b into x n. You have got x n. See the circuit, it is DSP circuit, signal flow graph also we find, x of n, it is getting multiplied by a factor b, it is a built-in constant and getting added with, if suppose y n is here, I will tell you what it is, we have done already this kind of thing.

Suppose this is y n and then there is a delay, built-in constant a. Let us say, if we assume that y n has been generated correctly till the previous clock y n minus 1. So, whatever I had at y n minus 1, at n minus 1 is clock, it is coming here now which is correct, y n minus 1, that times a gets added with b x n and new output is formed which is correct y n and this goes on. This is a circuit. What is the critical path?

Suppose this takes tm, this takes ta, this takes tm. So tm plus ta, you cannot go in this direction because there is a delay but this side, tm plus ta. Now this tm... the here I can have a cutset, only one edge, this is one sector, there is one sector, only one direction, no reverse direction, so this edge if I pick up, as I told you this entire thing will be cut into two half, this one half, this one half and so this is one edge only, no reverse edge.

So, I can add any amount of delay in this direction. So, I put a delay here, D. So, now this tm and ta they are not getting added because there is a delay, but still critical path will be tm plus ta. If you do like this, I understand if you remove this, forget about this yn now. I told you how to get y n, this is not serving any purpose here. So, you have D.

Now, see the dotted line, it is cutting this and this, this edge, this edge. They qualify to be cutset because if you remove them, on one hand you have this sector, other hand you have got just a multiplier, it is a single node. So, divided into two sub DFGs so which, therefore they form a cutset. This is one direction, opposite direction. In this direction you have a delay. If I remove the delay you can bring it here.

So, earlier I had tm plus ta, now there is a delay so the two times will not get added because of separation by delay. But now this has opened up. Earlier there was a delay. These two were separated. Now, there is no delay in this path. This is delay-free, so ta plus tm. So, critical path remains ta plus tm. Here the problem is I cannot in bring in a delay directly here because a loop, total loop delay is D.

I would have loved to have a delay here also, delay here also because these two times should not get added. So, this should be delayed. These two times should not get added in this direction so this should be delayed. If there is a delay here, there is no delay. If there is a delay here there is no delay. But, I want delay in both directions. Then the loop delay should be 2D. But I am giving only one D.

And as I told you in retiming I cannot bring in extra delay to a loop from outside. Total loop delay remains same before and after retiming. Before retiming I had D, either D here or D here, so after retiming also I have to have D. So, what do I do? If I still keep D, tm and ta get added. So, critical path does not go down. How to take care of it? I will carry out in the next class. Thank you very much.