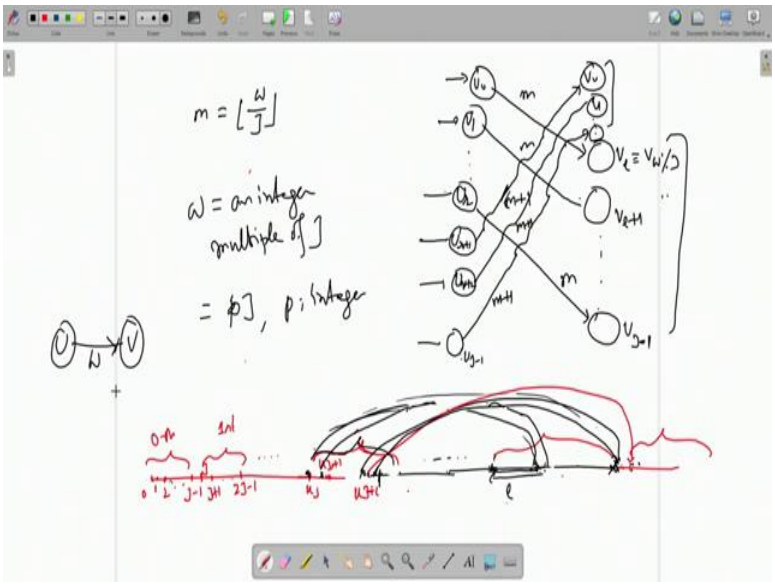
VLSI Signal Processing
Professor. Mrityunjoy Chakraborty
Department of Electronics and Electrical Communication Engineering,
Indian Institute of Technology Kharagpur.
Lecture 15
Retiming for Unfolding

(Refer Slide Time: 00:28)



So in the last class we started with unfolding that is Parallel Processing, we got such the example of an IIR filter. We will first give you the general idea of Parallel Processing. That if you are ready to increase the hardware by J times, then you can increase the input output clock also. Clock speed also by J or you can reduce the period by J, make it sorted, if the system is working and a clock capital T, your input output clock period can be T by J.

Where I took the example of an IIR filter and then I got such as the general case of unfolding a DFG where a typical sector of a DFG is like if you know 1 node U from VJH goes out to another node V and H may have some delay W, then how to unfold it, so you will be copied parallelly J times and V will be copied parallelly J times.

Now, one of the U, U 00 will, you know, I mean what is happening there just for a quick recap, J number of input data or incoming data parallelized, so in the 00th clock, in the 00th system clock, that is under 1 umbrella only, input clock 00, input clock 11, input data at 22, input data at 33, input data at J minus 11, so total J, they are parallelly coming, they are with parallel by a buffer and they come into the system at that the 00th system clock, system clock period is so much bigger because it is slow, so like that similarly first will be you know, having data at J, J plus 1 like that.

So, the leading data that is the 00th or J plus 00th or 2 J plus 00th et cetera will be processed by processor with index 0, like U 0 or V 0 like that. Similarly, data with index 1 will be processed by V 1, U 1 like that. Under this case if you take any particular, you know, any particular node Ur Urth copy that means or say i, ith copy, that means data is at KJ plus i, from KJ plus i I have to go to the right by W number of system to input clock because originally we were given something like this, these are the original stuff, these I cannot violate.

So, with respect to original input rate W number of W cycle of delay after being process by U, so after having process by Ui, because it is KJ plus i, if it is KJ plus 0 process by node 0, if it is KJ plus 1 process by node 1, they are parallel, so KJ plus i means ith input line will be given to node Ui and then if you start counting W number of input cycle, so 1, 2, 3 like that if you go for that to the right, you may be hitting here.

What is this point we worked out, that is instead of counting from here if you count from here then basically you would have jumped to the right not by W but W plus this much, this i, i plus W, so within i plus W how many blocks of J, so divide by J, cosset will be the number of cycle by which you have to go to the right, so that many system cycle delay, if cosset is 1 from here you are going to next.

So, 1 system cycle delay, if cosset is 0 no, in the same system cycle if cosset is 1 as I told is a next system cycle, if the cosset is 2 you are jumping to the right by 2 so 2 system cycles, because they are coming parallelly so this J input data or incoming data available together, next input data available together. Together means in the same system clock.

So, small W number of input cycle means you have to move here, so how many are system clocks. So i plus W you take divide by J, that is how many J are there take the cosset that will be the number of clock cycle, system clock cycle by which you have to go to the right there is up to delay then you hit here.

And once you hit here that would be here, once you hit here how much is this index, how much is this index? This is because if it is 0 then I have to give it to 0 th guy, because it handles suppose data with 0 index, like KJ plus 0 goes to U 0 or V 0 like that.

So, if it is here data from Ur, after so many cycles… system cycles delay would go move to V 0. So I find out what is this much this is nothing but the remainder, if the remainder is R or

say A. So from Ui, I go to VR we see some cycle delay equal to cosset of i plus W divide by J.

Then I there was a formula we derived last time and then if we see the implication of the formula. Suppose you start at a particular block you start here and you have to go to the right by W. So suppose you go to the right by W here. So many may be this is l and you go to the right by 1 next next. So after some system cycle delay that is if you divide i plus W i is 0 here is KJ plus 0 not KJ plus i.

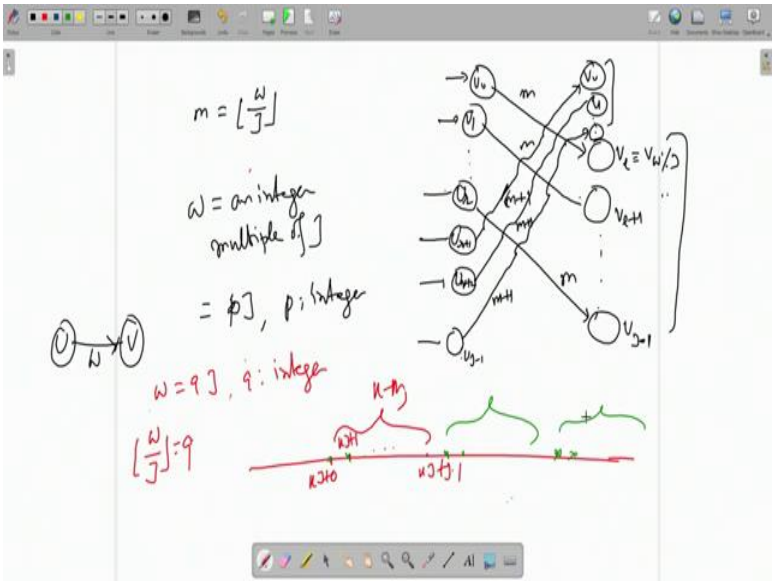
So, W by J and then take the cosset that is called M. So after M cycle you have to shift to the right you have to move to the right by M cycle and from here if you go, you are hitting this fellow this is l. That is why U 0 going to Vl, U would supposed to be V so I am going to V category from U category but which of the V be Vl and after how many system cycle delay M.

Then if you go on next this KJ plus 1 that is I am U 1 because index is 1, if I move to the right by 1 here also I go to the right by 1, so still under the same umbrella so still number of cycle by which I have to go to the right is M cosset will be M. That is, if it is not 0 plus W but 1 plus W by J cosset, cosset will be M but remainder goes up to l plus 1 that is why this goes to Vl plus 1 in the same delay it will go on till a time when I hit a point from where if I go to the right I hit here may be it is Ur may be it is here Ur.

There is a last point, still under the same umbrella so delay remains M there is r plus W by J cosset is still M but where is the last guy means J minus 1 th. So I have got this parallel lines with M delays, now if I go further to the right further to the rather enter the starting index of the next clock. So I need 1 more cycle to move further to the right. So delay will become M plus 1 and I will hit the 0 th index means I will go up to 0 th processor V 0, so from Ur r plus 1 I go to V 0 with M plus 1 the M plus 1.

Next also will be, if I go further to the right by 1 here also it will be going further to the right by 1 so from U 0 I will now go to V 0 I will go to V 1, if I go from Ur plus 1 to Ur plus 2 delay remains same M plus 1, dot dot dot dot finally is last point here will take will take me into last point here this is 1 category, this is 1 category. We got general structure of an unfolded edge.

(Refer Slide Time: 08:02)

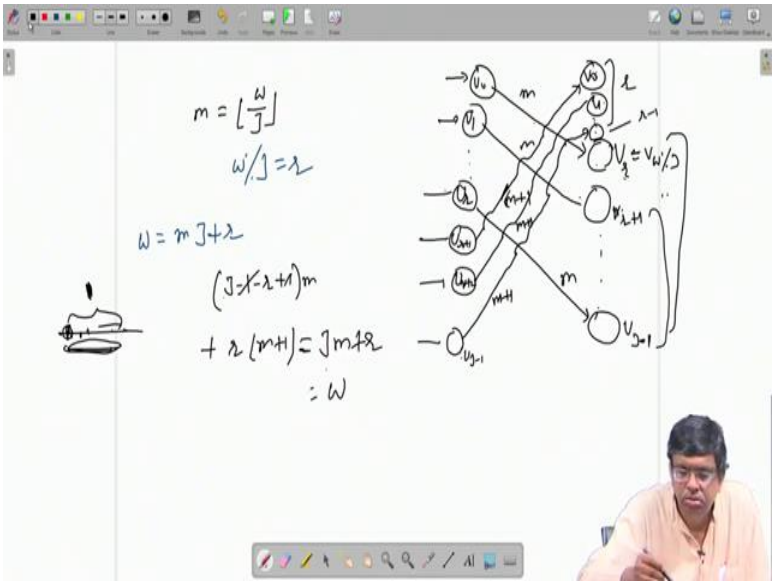


So after this further repeated or repetition of the explanation, I hope down it will be clear to you further. Suppose, as an examples this W is some integer multiple of J sub q times J q sub integer. Then what will be the structure? It will start with the timing diagram suppose you take the k th system clock that means it will be KJ plus 2, KJ plus 1 dot dot dot, KJ plus J minus 1. Now W by J cosset is q That if you start from here if you cover 1 J you are having the next these are next, it will start from 1, 2, 3 like that there is finally you will hit here.

So if q is 1 then W is J if you start at KJ plus 0 it go to the 0 th index or (())(08:58) making the next clock so 1 cycle delay but U 0 goes to the V 0 because 0 th to 0 th then first to first then second to second. So it will be horizontal line U 0 to V 0, U 1 to V 1, U 2 to V 2 all with 1 delay, you have got q delay go further like if q is 2 from here you go by 1 mode so you hit the next 1, so from KJ plus 0 to K plus 1 J plus 0 to K plus 2 into J plus 0 here.

So 2 system clock to the right 1, 2. So there will be U 0 will go to V 0 if we go to the right by 1. This also we go right by 1, so U1 1 will go to V1 U2 will go to V 2. So there will be horizontal lines now with delay 2 in general with delay q this will be structure. Just 1 side issue.

(Refer Slide Time: 09:53)



Another interesting thing which comes from this is this, so we are dividing W by J in general this will be the cosset M and remainder say W percentage J which is remainder or in general if W is less than J cosset will be 0 and remainder will be whole of W we all know if W is less than J if we divide W by J cosset will be 0 remainder will be whole of W.

Otherwise there will be some new cosset some remainder, so in general W then will be m times J cosset times divisor plus whatever will be the remainder. If m is 0 W becomes r this are all of us know, suppose this is the case the W is giving to be this and I just want a game, I want to play a game I want to find out total number of delays in all these unfolded areas.

So here I have got m, here I have got m , here I have got m, here I have got m dot dot dot up m so many m, m plus m plus m so that many and then m plus 1, m plus 1, m plus 1 so many m plus 1. So if we add all these m and m plus 1. So what is the total figure, I just want to count.

Now, here it goes to this node delay is m, here it goes to this node delay is m and then it goes up to the last node delay is m, how many I have in this category? This is Vl it is not just a minute. In fact I should it should be actually Vr now if I introduce then, because you start with 0 th index is a timing diagram you go to the right by m cycle then you must hit the r th point then only U 0 should go to Vr.

So if I call, that means remainder should be r that is if you divide W by J in cosset to m number of cycle to the right to the go and then you hit the starting point then further by the remainder if remainder is odd instead of l it should be r that is VW percent in J, VW percent

it is r that is why it is Vr that is why it should be V it should be Vl because l is replaced by r it will be Vr plus 1 and dot dot dot.

Just changing their notation a like so I have got r as the remainder, so how many I have here for which the edges have m delays I have got J minus 1 minus r, so that will be this many plus you have to take this Vr also so plus 1 J minus 1 minus r or r plus 1, r plus 2 up to J minus 1 you went.

So J minus 1 minus r will be this much and then you have to take this also that means how many I have got J minus 1 minus r and 1 more plus 1 right, J minus 1 minus r will give me this much and then I have to take this also into account because this also as corresponding is m delay that is plus 1.

1 more case, so how many such nodes here J minus 1 minus r this many plus 1. So for all of them I have got delay m so that many m and the remaining edges have got m plus 1 delay how many of them total I have got J total I have got J and here I have got 0 1 up to r minus 1 because here it is r.

So 0, 1, 2 up to 1 r minus 1. So total is r here you can see these 2 cancel so J minus r so here I have got J minus r total J so remaining here is obviously r and if you count this is U 0, U r dot dot dot U r minus 1 so total is r. So r times how many delay for this edges not m but m plus 1 if I see what are the total number of delay when they are added in all the edges after unfolding what is that?

So minus mr and plus mr cancels so what you have is Jm plus r that is all which is nothing but this mJ plus r which is W this is very interesting, that if the original delay interup of the original input cycle in that edge is W.

Now, obviously delays in each edge going down because you are dividing by J undertaking the cosset. So as a numbered m is must may be smaller than a W this m smaller than W m plus 1 smaller than W, m plus 1 smaller than W. So number of delay number of delay, delay counts not intercept time delay counts there is a figure wise m is much less than W, m is less than W m is less than W but when you add them they add up to the total W there is total W figure gives distribute among (())(15:18) this as smaller figures m, m, m, m then m plus 1 m plus 1 m plus 1 m plus 1, when added figures becomes W.

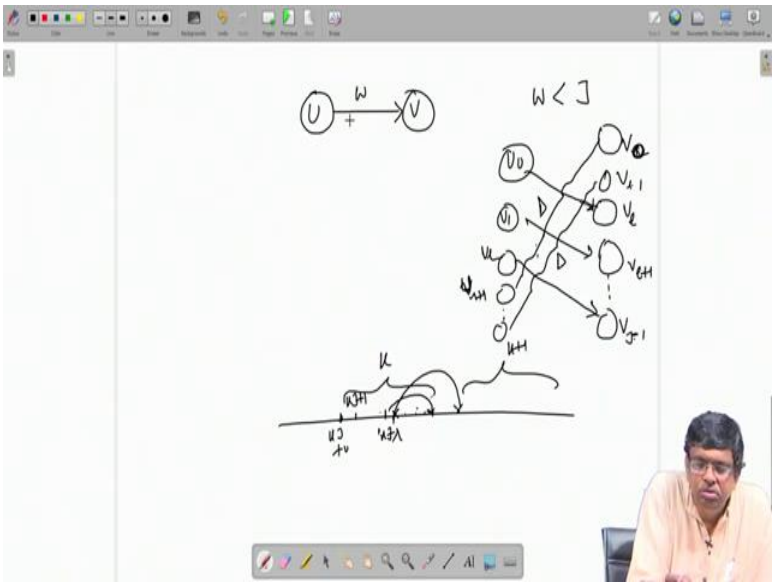
But remember W was with respect to the input clock original input cycle which is faster. So here that would have been 1 input cycle so this is faster. So take W of them that many delays.

But now when I say m, m is with respect to the system cycle. System cycle means J number of input cycle if you take J such that this is 1 system cycle this much. This is 1 system clock period.

So in terms of time it is same in terms because it is 1, earlier I had got J number of input clocks. Now J all packed together and the data here are parallelly available in 1 system clock. So figure has g1 down from J to 1 but in terms of time original clock had only this much width now I have got so much width. So, in terms of time or clock period duration each of this unfolded that it has longer bigger clock period than the original.

We in terms of the number of such clock periods obviously it is glaze m number of clock such slower clock period that is less than W, so as I figured this less than W and interestingly if I guess add up the figures only it turns out to be W.

(Refer Slide Time: 17:03)



Another interesting thing, suppose I have got U V and there is a W delay and W is W number of input cycles and W is less than the unfolding factor J, J is an unfolding factor and W is not very high this is less than J then what will happen in terms of system cycle you can see if I am at a particular system cycle Kth for all the data are parallel, so this is your K J plus 0 KJ plus 1 plus dot dot dot dot some point.

And then next 1 is K plus 1, so now if I started KJ plus 0 that is if I have got U 0 and if I go to the right by W number of input cycles so 1, 2, 3. So W is if W is equal to J from KJ plus 0 I go to this point (())(18:00). KJ plus 0 if you start counting J you hit here, so you go to the next cycle.

But since W is less than J if I count W to the right I would not reach here, I will be rather here, I will be rather here or if W is less I will be here or here or here or here. So from here I will be going to some edge may be Vl but within the same system clock, so 0 system delay 0 system cycle delay and as you know this will continue for a while this will continue for a while there will reach the last guy. VJ minus 1 and then if you go further to the right it will hit.

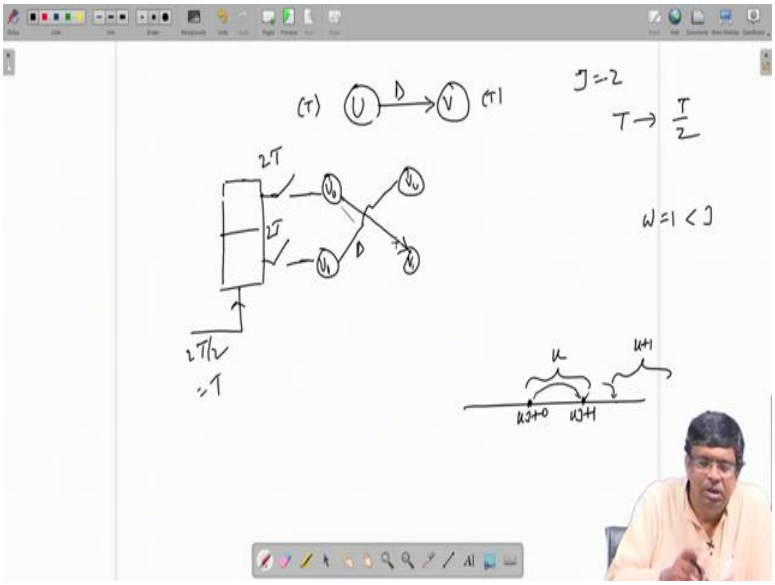
So here, it means if it is say Vr that is we have got 2 KJ plus r from here if you go to the right by W you hit the last point from here you went here, here you went here, here you went here like that but you have now point KJ plus r. So node is Vr that takes that this data but data is parallel. From here if you go to the right by W you are still within the cycle same system cycle but in the last point that is why here.

So, all these edges have no delay and in this direction I have seen the general structure if it is m this m plus 1, if it is 0 it is 0 plus 1 that is 1 delay that you can see here also. If you go for that to the right you hit the starting of next guy. That is Vr plus 1 will take you to sorry, Ur plus 1 will take you to Vr V 0 at (())(19:55) 1 delay, and then like that dot dot dot, Vl minus 1 still 1 delay there is a complete all the points you hit this.

So interesting thing is that, originally I have W number of delays and W suppose is not 0 greater than equal to 1. So really there is at least 1 cycle delay or more than that, but after unfolding certain edges arise these edges in this direction which are delay free no system delay no delay at all so they are the nodes carving the same cycle.

Nodes raises in other directions are delay but in 1 direction these happens when W is less than J, whereas the delay is less unfolding factor is more, so more data parallel lines so that is why some delay few paths emerge this another interesting thing. But this has an implication now, I will show your implication.

(Refer Slide Time: 20:58)



Suppose I had a process is U and V, this takes time T this takes time T. So critical path is T plus T into T I mean to minimize the critical part (())(21:13) timing here I put a delay I put a delay. So now a critical path is T because in one cycle I do the job here next cycle only that that output is required here it takes a time T to do the operation on that, that time it takes time T to process next data, I like that we have been discussing this again again again. So now critical path is T.

But I say no, I am still greedy I want to make it T by 2 but nothing more can be d1 you cannot go inside nodes and you know do anything that is suppose not possible. So even if you increase D to 2 D 3 D 4 D anything it does not change the critical path, critical path is T you have to be happy with this. If you are just doing things this way by pipelining.

But I say no, I still want it to be T by 2. So if we have to be T by 2 then what have to do is this, I take a buffer J is 2 J is 2 because from T, I want to go to T by 2, T by J here T by 3. So J is 2 means I need a buffer of 2 chambers, as before X of 0 X of 1 then X of 2 X of 3 even odd even odd even odd that combination comes.

We have got, so we just closing at a period of T. Now U will be U 0 V 1, it will take X 0 then X 1 that he have. Next time it will take X 2 that time X 3, next time it will take X 4 that time X 5 like that. This we have, I have taken up this example in that IIR filter case so I have not reading it, and rate here is T by 2.

And now unfolding, so here V also becomes V 0 V 1. Now if you see the timing diagram the general block su ppose you have got 2 data, KJ these are Kth block KJ plus 0 KJ plus 1 these data goes to U 0 KJ plus 1 th original data goes to V 0 U 1, U 0 V 1 fine.

So, after being processed I know that after U, I have to give one input cycle delay, one input cycle delay remember here it is 1 W is 1 but J is 2. So W is less than 1 W is less than J, here W 1 means less than J because J is 2. So it is a case I consider in the previous page. So if we start from KJ plus 0 you have to go to the right by only 1 because there is a correct, there is a actual situation given to me, I may no implement in my ways but actual thing given is output of U should be delayed by 1 original input cycle then given to V for processing.

So, I started if I started KJ plus 0 that data is processed by U, U is U 0 here. So I have to go to the right by 1 input cycle from here. So from KJ plus 0 I go to KJ plus 1 within the same system cycle. So no delays is required I give it from index 0 to index 1 so it comes here.

And from here if I go to the right I go to the next cycle. So 1 delay and from index 1 to index 0. So U 1 goes to V 0 (())(24:46) but with 1 delay. Now you see originally I had a delayed in the edge so the 2 nodes U and V there timings were not getting added because they are not coming in the same cycle. There is 1 cycle it does the job then gives it here in a, and then it processes it here is the next cycle, then it does not depend on the same cycle here.

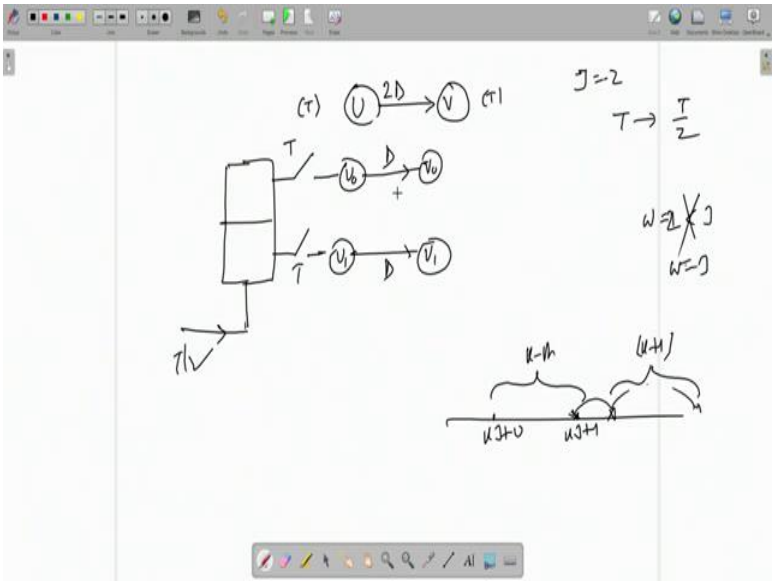
So it is doing job it is doing job independently, previously computed job is used here, so it does not we have to wait till in the current cycle till it finishes it is job. That is why critical path was just T, but here you see this edge has got 1 delay but as I told to you W is less than J as is the case here, edges in this downward direction become delay free so it is delay free which means this time T and time T they get added not for this edge but for this edge because I have to take the edge which takes maximum time I have to go by that because I have to give time for that also to be complicate not only the other 1.

So this edge will require T plus T 2 T. T plus T 2 T. So system has to system clock cannot be get shorten then 2 T, then we instead of 2 T it will become 2 T and then 2 T by 2 means T. So what have I achieved, I have achieved nothing input clock period still remains T, I had first retained it was giving me critical path T of input clock was T only period was T, I say no I want to make it T by 2 somebody told ok no unfolding. So I unfold with 2 and got something like this but I find its critical path because the 2 nodes and now joining into the same cycle the delay has g1 there is no separating delay, so the 2 times get added T plus T 2 T therefore system clock cannot be shorter than 2 T so it will 2 T this is T.

So input period remains T but unnecessarily I have increased my hardware earlier I have 1 U node I have got 2 U nodes, earlier I have 1 V node now I have got 2 V nodes. So I am requiring double the hard work double the power without getting anything. So give me a minute, let me put my mobile on vibrater mode.

So, I did not get anything. So there is no point where you can say that what is the advantage of being unfolding. Now you see while doing retiming originally I have no delay then I retimed and put 1 delay and then went ahead with this.

(Refer Slide Time: 27:34)



So I do something more there, if instead of 1 D. I give 2 D even then critical path here is T so input clock cannot be shorter than T clock period because only these D or 2 D or 3 D that does not change anything. This is working in 1 cycle taking T around of time it is giving in another cycle here taking T around of time, we have d1. So this is critical path remains T but now after doing making 2 D here after having a 2 D here if I do unfolding by same factor 2.

So I will have, now you see, if I go back to that timing diagram if we suppose Kth. Now here and here, now only 2 point KJ plus 1 KJ plus 1 and from KJ plus 0 I have to go to the right by now 2 not 1. Now it is W equals to 2, so this is not correct W equal to J now.

So if we go the right by 2, 1 another 1, so I am here, so K plus 1 th which means I have to delay this output by 1 system cycle and from 0 th, I go to 0 th, so it will go to V 0. But there will be delay. Similarly this one will go to the right by 1 and then 1 more because it is 2, so from the index 1 to index 1 U 1 will go to V 1 but same system cycle.

So now, both these edges have critical path I mean this system has critical path T only not 2 T because this edge also having a delay this also having a delay. So the 2 times of the 2 nodes T and T they do not get added. So critical path remains T, so system can (())(29:41) will work at clock period of T so if it is T it will be T by 2.

So, see just by changing the retiming before going into unfolding we can get our desired result. This is very important that I am unfolding alone may not give you the result because unfolding has a problem that delay count for edge goes down, that m or is a cosset of W by J revision. m number of delay m plus 1 number so delay count goes down and in W is not high J is high some delay free paths emerge, some delay free paths emerge in 1 direction it will delay free path.

So originally, edge was delayed but now new edges come up which is come up which is delay free, so instead while trying to achieve higher speed I (())(30:32) problem had critical path increases, so I cannot really increase the system clock and therefore I cannot get the confront edge of the input clock, because unfolding has a problem that it always leads to reduction in the number of delays in the unfolded edges.

And if W is less original delay W is less than J then each edge then each edge will have delay, delay free I mean some of the edges will be delay free and some of the edges will delay 1 if W is less than J and so critical path will get increased. The computation time of the nodes involve will get added. That is why it is important that you do retiming before and I mean suitably so that after unfolding also no new delay free paths emerge.

So, these are some theory actually that we will not cover but those who are interested they can read something from paradies book, it is more it is pose more like this that after unfolding if we want to have a particular critical path that is the unfolded system should not have critical path greater than some value say see.

Then how should I do the retiming of the original DMG, so that after retiming when I unfold the critical path of the unfolded system meant meats that requirement that is not exceeding amount see. These are the theory, this if you interested you can see. Thank you very much in the next term I will continue from here.