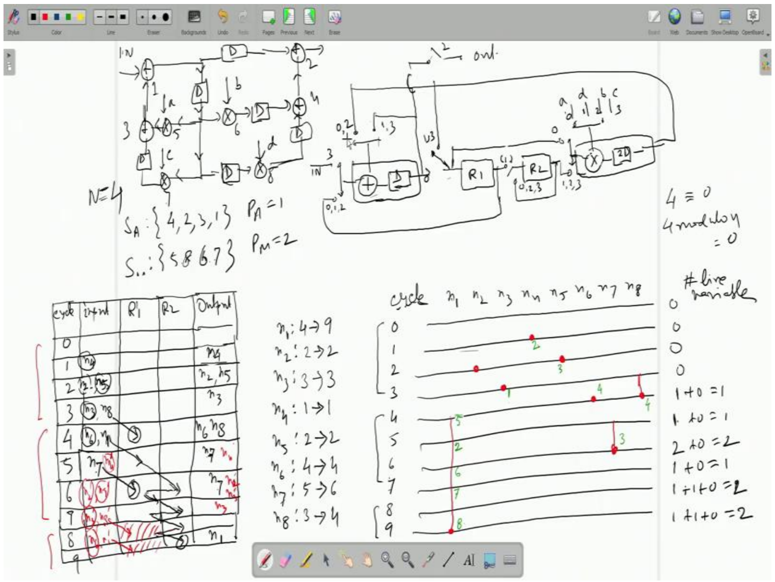
VLSI Signal Processing
Professor. Mrityunjoy Chakraborty
Department of Electronics and Electrical Communication Engineering
Indian Institute of Technology, Kharagpur
Lecture – 28
Delay Folded Realization of a Digital Filter

With the great difficulty I have been able to draw this circuit which is complicated.

(Refer Slide Time: 0:28)



$S_A : \{4, 2, 3, 1\}$ $P_A = 1$

$S_M : \{5\ 8\ 6\ 7\}$ $P_M = 2$

cycle	input	R1	R2	Output
0				
1	n_4			n_4
2	n_2, n_5			n_2, n_5
3	n_3, n_8			n_3
4	n_6, n_1	n_8		$n_6\ n_8$
5	n_7 [illegible]			n_7 [illegible]
6	[illegible]	n_1		n_7 [illegible]
7	[illegible]			n_3
8	[illegible]		n_1	n_1
9				

$n_1 : 4 \to 9$
$n_2 : 2 \to 2$
$n_3 : 3 \to 3$
$n_4 : 1 \to 1$
$n_5 : 2 \to 2$
$n_6 : 4 \to 4$
$n_7 : 5 \to 6$
$n_8 : 3 \to 4$

cycle	n_1	n_2	n_3	n_4	n_5	n_6	n_7	n_8	# live variable
0									0
1									0
2									0
3									0
4									1+0 =1
5									1+0 =1
6									2+0 =2
7									1+0=1
8									1+1+0 =2
9									1+1+0 =2

The board is not also very good, it is very disciplined. Anyway let us start this. I first have drawn that adder, this adder which have you need plus 1 pipeline latch which is sold separately. One input which is switch, another input which is switched. We have drawn one adder one with input switched output and R2 with input switched output. Then a multiplier that has, this is multiplier, this not be this not be an adder.

By mistake I have drawn as adder which is actually a multiplier. This multiplier one input which is built in constants a, b, c, d that you can see; a is for a multiplier 5. Multiplier 5 is scheduled is 0, that schedule 0 it touches a and then takes x as multiplier 5. This input as a, then next is 8, 8 is this fellow d, this multiplier which is built in constant d and schedule is 1. So, the next schedule is 1 that time it takes d. Then schedule 2; schedule 2 is 6, multiplier 6, multiplier 6 has constant b.

So, b is at schedule 2 and lastly 7, multiplier 7; multiplier 7 has constant c, its schedule is 3; that schedule is 3. Let us see how we brought this circuit. Let us start with n1. In n1 you see n1 is born here, you cycle 4 and cycle 4 in modulo notation were I mean part is uses modulo notation. So, 4 is equivalent to 0 because 4 modulo modulo 4 is 0. Why modulo because every 4 cycle cost to a block. Say if it is a 5, then 4 cycle is done at one more. Say it is the cycle of next block.

So, it is a modulo conversion you rested yourself into this either cycle 0, 1, 2 or 3. It could be for the next block or next to next block it does not matter. If you there is notation that this book by Parhee follows just I am sticking to that here. But if you are not not comfortable you can write 4 as well instead of calling it 0. But I am following this book, so that there is no confusion instead of 4, I will write as 0. So, at cycle 0 which is equivalent to 4, n1 is born. This n1 as per this diagram, n1 born it will go R1 input.

So, therefore, this switch comes here equals to R1 input. Now, one thing you see as n1 is born even the cycle 4 itself, there there is without big store it goes 5. Device 5 device 5 means multiplier 5, input of multiplier 5. That is why there is a parallel path directly going to multiplier. And this multiplier input will touch it in the same cycle 4 which is equivalent to 0. So, it will directly go here and go to the multiplier.

This multiplier in the cycle 0 that is, cycle 4 will work like will work for this, okay multiplier 5. Cycle 4 it is out and 4 to 5 here the same cycle it will be going there. So, cycle 4 it is born which is 0 and is a same cycle it goes to multiplier input, multiplier 5. Now, after one cycle in cycle 5, there is after being stored by device stop 1 R1. This is u stop 2, 2 means device 2, device 2 is an adder. So, there is an R1 output this comes to here and the adder input.

That time the adder is working for adder 2 and what cycle, cycle 5. Cycle 5 is equivalently 5 modulo 4 is 1, so take this 1. So, this is touching at 1. Otherwise, you look at n7, n8, n7, n8. I will complete this journey of n1 for the timing I have gone from cycle 4 to cycle 5. It has come here and cycle 4 to 4 itself going to multiplier 5. So, that put have done. Now, consider this n8; n8 is born at the output of multiplier, multiplier 8 and it goes to adder 4 after one cycle.

So, n8 will be coming from multiplier output. So, from this output line, then again stored by one cycle, so stored so that it will touch this here at what cycle? Cycle 3. Cycle 3 n8 is born, so if touches this, it will take the multiplier output which is this fellow n8 cycle 3. It will store in RO1 and then consumed. But after being consumed where will it go to? 4, 4 is an adder. So, that is why it is going here to this input again; that time it is working at the adder 4, which schedule, which cycle? Cycle 4; which is equivalently 0 for modulo 4 is 0 that is this 0 standing for.

And consider again n7 basically I am considering storage by 1; here is storage by 1, storage by 1, storage by 1. 4 to 5, 5 to 6 and 3 to 4. This storage by 1 is done by RO1. I am considering that storage by 1 cycle. So, now again n7 born here in 5 and it goes is stored by R1 and then consumed, it goes to 3. 3 is an adder, so again it will go to adder and it is 1 in 5. That is why and 5 modulo 4 equivalently 1. That is why its multiplier output is coming here; this R1 input touching there at 1, that is 5 equivalently, and the output is coming here.

In the next cycle what is the cycle? Where 5 it is born 6. 6 modulo 4 is 2; that is this 2. So, this n7, n8 done and n1 I went from 4 to 5. Now, consider 4 to 5 so 4 to 4, that's why it is going to multiplier 5 done. Then 4 to 5; R1 output going to adder 2 done. Then now one more cycle 4 to 6; now what is happening look at this table. 4 it goes to the input of R1, so its cycle 5 it comes to output of R1 and it stored in 2. That is giving to R2. So, R2 cycle 6, it will come to the output of R2.

So, if it is at 4, this fellow will touch it which is 1, 1 means 5. But cycle 5 it will touch, it will come here to R2 output; at what cycle? Cycle 6. After that what is happening is this, cycle 6 and it is fed back to the input of R2 itself it is fed back. It is cycle 6; 6 is equivalently 2; 6 modulo 4 is 2, then it is stored by R2 itself. It goes to the output of R2 at cycle 7. Cycle 7 means cycle 7 modulo 4 that is 3. So, at 3 it comes here and then again it stored 8.

Cycle 7, so 7 it comes here, then 8 cycle 8 is equivalently 0, 8 modulo 4 is 0. 0 again it is stored and finally this is coming at 9, 9 is equivalently 1. And during this journey what is happening? You see after stored by R1 then R2 it goes to 6. 6 means multiplier, so it goes to this multiplier at what cycle? It was cycle 5 that is 1. After here cycle 6 it goes to 6. So, 6 is equivalently 2, so it goes to multiplier at input 2. Then it is stored 6 and cycle 6 it is stored cycle 6 it is 2.

So, one more cycle, cycle 7 in cycle 7 it is used up by again device 7. Device 7 is multiplier, so that is why it is coming here and 7 is equivalently 3. 7 modulo 4 is 3; so it is touched here at 3. Then at 7 again it is stored. 7 means 3 so 3 it is stored, it comes out at 8. 8 I do not use it anywhere; but 8 is means 8 modulo 4 is 0; so 0 it comes again, then it is 9. At 9 it is used up by 8, device, 8 is a multiplier and 9 modulo 4 is 1; that is why it is touching here at 1.

So, this is the 1, 2, 3. This takes care of journey of n1, n7, n8 done. Now, consider n2. I will come later because n2 does not do anything; n2 output is taken out, that is output. Start with n3; n3 born in cycle 3 and dead in cycle 3. Cycle 3, so n3 is the output of adder; so the output of adder goes out at cycle 3 and that time what does it go to? It goes to adder 1. So, it goes to the other input of adder; so from here it goes to the other input of adder at cycle 3 and I am touching it.

Then comes n4. n4 also is the adder output; n4 adder output born in cycle 1 and it consumedly cycle 1. What does it go to? Device 2 which is an adder; so it goes to the other input of the adder at cycle 2, cycle 1. 2 is the number cycle 1, so again I am touching it at 1. So, 1 corresponds to this n4 being born in cycle 1 and also consumed in cycle 1, that is n4 going to n2, adder 4 output going to adder 2 input. And similarly, consider okay consider n2. n2 is the output of adder 2, adder 2 is schedule is 1 and then 1 internally it will schedule to 2.

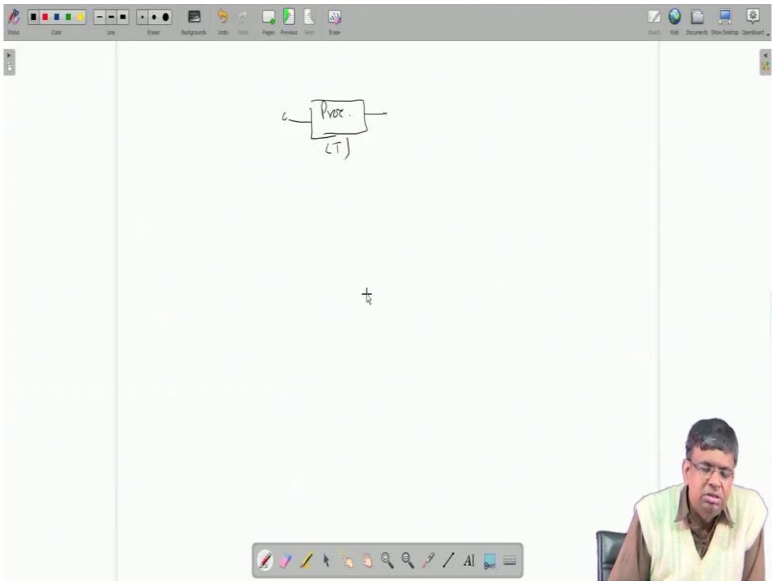
And schedule 2 it will schedule you to the output. So, this adder output I am touching here at schedule 2. I am taking it to the output, so n3, n2, n4 done. n5, n6. N5 and n6 coming out of multiplier so multiplier output. Where is n5 going to? n5 going to 3, who is 3? Adder. So, n5 will be going to born and dead in same cycle, no need to push it through this registered chain. No need to store; but n5 will be going to adder; adder 3 so other input of the adder. At what cycle? Cycle 2; so n5 directly going to the other input here cycle 2.

N6 again going to adder 4 device 4, device 4 is adder so again it will also go to the other input of adder. But at what cycle? Cycle 4 which is equivalently 0, 4 modulo 4 is 0. So, output is going here at 0. So, this completes the circuit; both hardware optimized that is the multiplier and the adder optimized because I am using only one adder, one multiplier. At D optimized only two delays R1 and R2, earlier I had about 6 7 delays.

Switches you can say that switches are more switches are brought in here. So, they may use of hardware but the point is, switches are you know just I mean (())(13:09-13:13) switches. And here n is 4 but n could have been a big circuit or n could have been hundred. Then also number of switches could have been same so, switches do not add complexity so that much. So, here we stop for the folding part.

We have learned the folding of hardware units, then we have done some theory, retiming for folding, cuts and retiming about folding, so that no edge turns out to be negative. The new head for delay optimization by folding. We consider lifetime analysis lifetime chart, then forward backward data location and finally construction of the circuits. We took some examples and finally work out the same thing for a digital filter. So, I think we have given enough coverage of folding. We now say for the next topic.

(Refer Slide Time: 14:19)

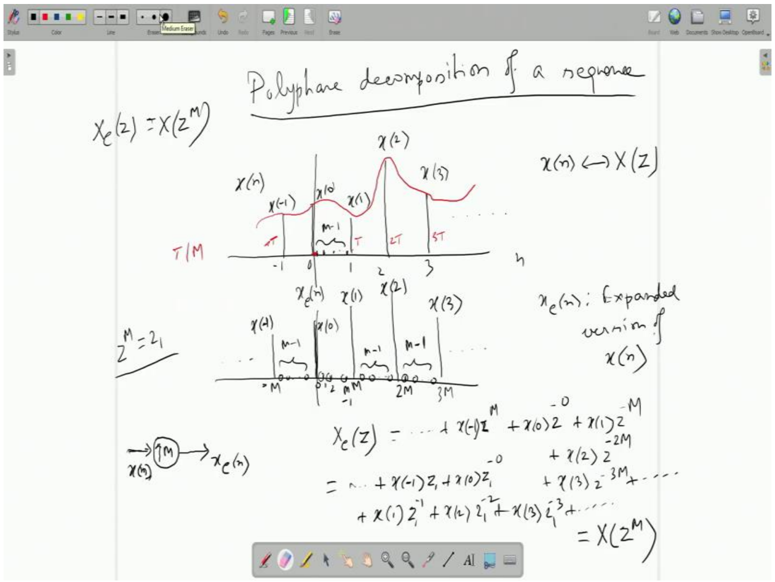


I will be going little back to again little back going back little to unfolding again. We have seen in that case of folding, if originally one process is given to you some jobs, some computers are it was taking T amount of time because critical part was T, after doing whatever possible retiming you could do. Then what will do it will make it parallel. If you want to bring down input and output, if you want to bring up the input/output rate by a factor of j, that is reduce the period by a factor of j.

Then I need to persistent in clock or move up by that rate. So, I need to j copies in parallel. So, that has one disadvantage that, whatever hardware he takes that gets multiplied by j, whatever power it takes that it consumes that get multiplied by j. In general that is true; but in the case of certain circuits like you know I mean, if I are digital filters we can show some techniques. By which we will see that, hardware cost will not go up by a factor of j.

It will go up by some fraction of j. We can have some considerable hardware saving, though speed wise there is no loss. Input-output speed will still go up by a factor j. Now, to do these we need to have some idea about what is called polyphase decomposition and little bit about multi-rate signal processing. Very little which will do here.

(Refer Slide Time: 15:38)



Polyphase decomposition of a sequence

$X_e(z) = X(z^M)$

$x(n) \longleftrightarrow X(Z)$

$x_e(n)$: Expanded version of $x(n)$

$z^M = z_1$

$$X_e(z) = \cdots + x(-1)z^{M} + x(0)z^{-0} + x(1)z^{-M} + x(2)z^{-2M} + x(3)z^{-3M} + \cdots$$

$$= \cdots + x(-1)z_1 + x(0)z_1^{-0} + x(1)z_1^{-1} + x(2)z_1^{-2} + x(3)z_1^{-3} + \cdots$$

$$= X(z^M)$$

Suppose, I have got xn, one sequence. This is 0th sample, this is minus 1, x minus 1. Xn has a jth transform with some appropriate ROC X (Z), this is fine. What I do is this now, I bringing additional points here, M minus 1 points. So, what I do? I have the same sequence, same values but it is 0, with I have got new points 0, 1, 2 dot-dot-dot up to some M minus 1. M is some chosen integer minimum value 1, minimum value 2. So, at all these points I use to do 0 value, so this will be M.

At Mth point what I have? The x1 sample, earlier it was standing at point number one, now it is standing at point number M something point M. Then again I introduce zeros, M minus 1 zeros. So, Mth point, then M minus 1 so there will be 2 Mth points. Who is standing, same x 2, then again 0, 0, 0, 0, 0, M minus 1 zeros, M is an integer of your choice greater that or equal to 2. So, that 2 minus 1 is 1 that is at least 1 is there, otherwise this exercise has no meaning.

So, between every pair of samples, I introduce zeros dot-dot-dot-dot. This sequence I call xe subscript e n; e stands for expanded. xe n expanded version. So, e stands for expanded. We denote like this x n we draw it again, still its not coming, the problem with this board. Very solid alignment problem on this board. These are notations is called the expansions. Actually, if you want to do in real time that means earlier may be you have heard an analogue wave form, which was sampled may be an analogue wave form like this.

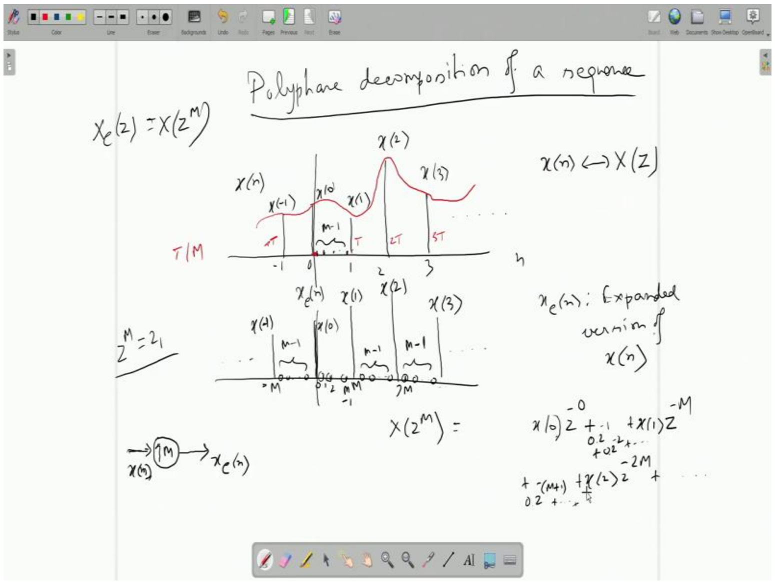
We sampled here, sampled here so these are the sampling points. 0 may be capital T in terms of time T, 2T, 3T minus T or like that. Now, you are creating intermediate sampling points, for sampling period now you are solving this much. T by M but you do not have the samples; you do not have these values, you do not know because it came and you took samples and single has gone. So, I am just putting 0 values. I am not approximating this wave form, I am just putting 0 values. But, from this analogy you can see that xe n in terms of real time operation can be viewed as, a Foster sequence.

Foster by a factor capital M, it is a period has gone down from capital T to T by M. T by M so it is a Foster sequence and Foster by a factor capital M. Then what is jth transform of Xe? This is nothing but jth transform of this. Now, this fellow is x 0, if you take jth transform, write jth transform equation, x 0 z to the power minus 0 because 0 is the index, then these zeros do not mean anything. Then next sample is x 1 but it is z to the power minus M because sampling point has changed now, index has changed.

Then again zeros do not mean anything, then 2 Mth points then you have got x 2. Again zeros do not mean anything next is the 3M. So, there x 3 z to the power minus 3M plus dot-dot-dot-dot-dot. On this side x minus 1 Z to the power minus this is minus M down; so minus and minus plus. So, z to the power M and dot-dot-dot-dot-dot. Suppose z to the power M, I write as Z1. So, this is nothing but x minus 1 Z1 plus x 0 z to the power minus 0 and Z1 to the power minus 0, they are same.

Z1 to the power minus 1plus x 2 Z1 to the power minus 2, x 3 Z1 to the power minus 3 and dot-dot-dot-dot-dot. Which is nothing but original Z transform but evaluated at 0, 1. Just you see this is nothing but original Z transform evaluated at Z1. What is Z1? Z to the power M. So, I write this result here, Xe z. ROC here will be a part of original ROC, this you can find out. I will not get into that; because this is not a class of DSP.

(Refer Slide Time: 24:16)


Polyphase decomposition of a sequence
$x_e(z) = X(z^M)$
$x(n) \longleftrightarrow X(z)$
$x(n)$
$x(-1)$
$x(0)$
$x(1)$
$x(2)$
$x(3)$
M-1
T/M
T
2T
3T
-1
0
1
2
3
n
$x_e(n)$
$x(-1)$
$x(0)$
$x(1)$
$x(2)$
$x(3)$
M-1
M-1
M-1
-M
0 1 2
M
M-1
2M
$z^M = z_1$
$x_e(n)$: Expanded version of $x(n)$
$X(z^M) =$
$x(0)z^{-0} + 0.z^{-1} + 0.z^{-2} + \cdots + x(1)z^{-M} + 0.z^{-(M+1)} + \cdots + x(2)z^{-2M} + \cdots$
$x(n)$
↑M
$x_e(n)$

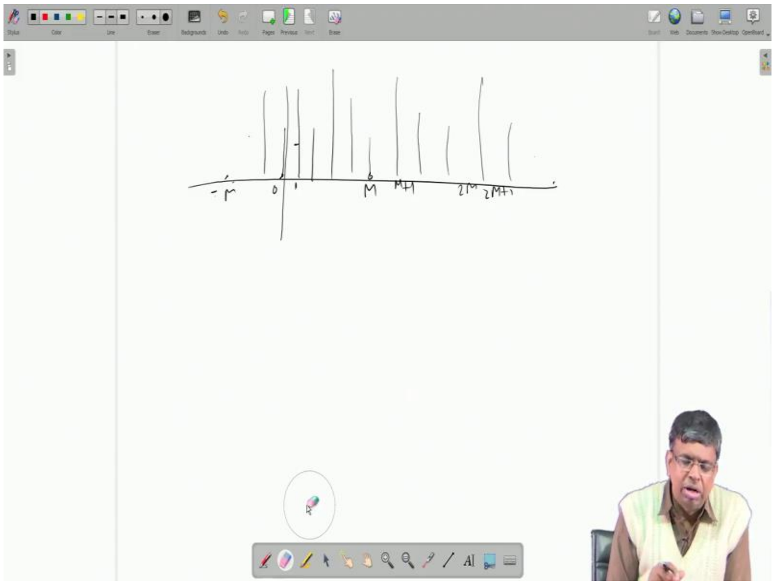
This expression. Equivalently, if I give you if somebody has given me X z. In X z, z has replaced by z to the power M. That is X z to the power M, this is nothing but x 0 z to the power minus 0, which is z to the power M minus 0, which is same as z to the power minus 0. So, this I do not change. Then x 1 is z to the power M, x 1 instead of z, z to the power M so this.

Then again x 2 z to the power minus 2M dot-dot-dot. So, that means I can write it as, as though there are terms here with 0 values, 0 into z to the power minus 1, 0 into z to the power minus 2 and dot-dot-dot-dot-dot. So, between x 0 z to the power minus 0 and x 1 z to the power minus M, I can assume M minus 1 terms when value 0 at z to the power minus 1, z to the power minus 2 and all that.

Similarly, between x 1 z to the power minus M and x 2 z to the power minus 2M; here again I can assume 0 value into z to the power something whatever, what is that? Minus m plus 1 plus so-so all 0 values. So, it is nothing but z transform of sequence of this kind that is giving xn if I plugging zeros like this M minus 1 zeros between every pair of samples. Then take the z transform that become same as capital X instead of z; z to the power M.

The moment you replace z by z to the power M that means in time domain between every pair of samples. You have band of zeros coming how many? M minus 1, this about expansion.

(Refer Slide Time: 26:34)



Using these, we develop the idea of polyphase decomposition, but I think that will take more time. I just got 5 minutes left. I would rather not do that here. But, what I am going to say is this that this sample, this wave form you know any analogue wave form. When you take samples or may be we do not use analogue wave form at all. What I will do I am just giving you the philosophy because in such a short time, I cannot cover it today.

Suppose you are giving signal a sequence, I will relate certain symbols certain samples at 0th point, Mth point 2 Mth, 3 Mth, minus Mth and all that just pick up them. That will be called as polyphase factor 0. Then I will pick up sample at 1, sample at M plus 1, simply that 2M plus 1 and like that. And form a sequence that will be called polyphase factor 1. Similarly, polyphase factor 2, so you can see that maximally available polyphase factor M minus 1.

Using them I would be able to get back by sequence, that is what I will show and that is called polyphase decomposition of a sequence. Each of these polyphase components will be very handy for me in the hardware in the unfolded version of unfolded realization of a FIR filter. FIR filter have a finite length impulse response, when I try to implement it in hardware. If I do ordinary unfolding I mean hardware cost will go up j times.

But if I follow certain trick tricks we use this polyphase decomposition; it will not go up by a factor of j time, but by certain fraction when j by 2, j by 3 like that. This very interesting thing you should, it is very involved things get fully into it, but at least j equal to 2 case and j equal to 3 cases will consider. So, that is all about next day class. Thank you very much.