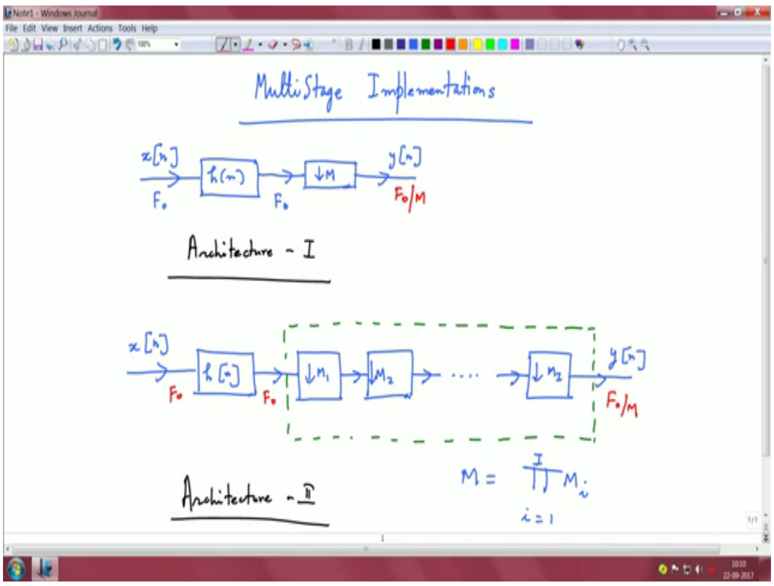
Mathematical Methods and Techniques in Signal Processing - I
Prof. Shayan Srinivasa Garani
Department of Electronic Systems Engineering
Indian Institute of Science, Bangalore

Lecture - 45
Multistage filter design

Welcome all of you to the class. So, in the last lecture, we discussed how one can realize efficient implementations using Polyphase representation, noble identities, the identity on interconnected systems etcetera, and then, using all these tricks how we can realize efficient architectures for rational sampling rate conversion. And one of the important considerations in the design of high decimation filters is the following issue. Suppose, I give you, you know a decimation rate of a huge factor, say 100 or 1000 to be realized.

Some questions that naturally arise to us is, do we just do some filtering followed by down sampling by F, a huge factor say, 100 or 1000 or do you want to break this into different components, where I have filtering followed by down sampling by 10 and then, another down sampling by 10 and so on, or may be filtering, then, down sampling by ten. Another filtering down sampling by 10 and what are the considerations right? So, this is, some thought one would get, but let us go a little bit into the details of these architectures and see, if there is any benefit of multistage decimation or interval in my expansion or interpolation accordingly.

(Refer Slide Time: 01:59)

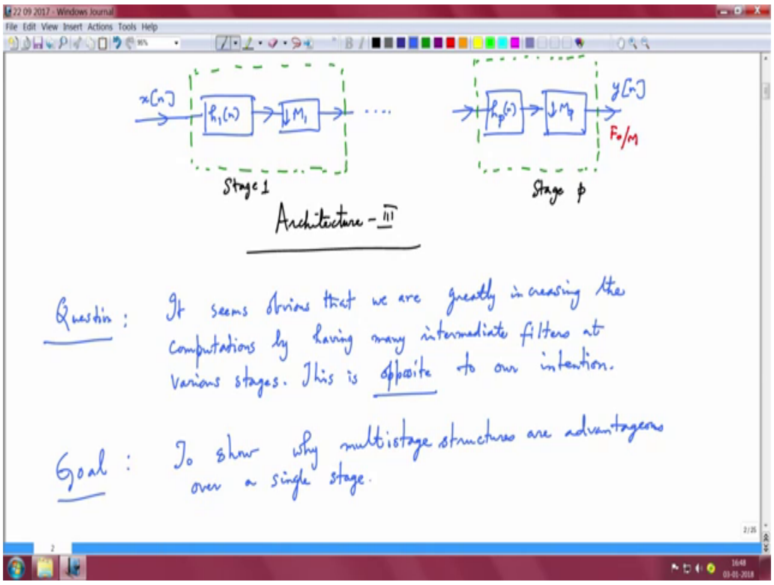


So, let us start with some schematics here. I have a sample discrete time signal, let us say the sampling frequency at the input is F naught, and at this stage we are filtering, so we are preserving the sampling rate. We down sample this by a rate M and we have the signal here at a rate F naught upon M, and I say this is architecture 1. This is a normal thing, I mean I have a discrete time signal, I filter it with some filter typically a low pass filter with some cutoff and some transition band, and then you down sample this by a factor of m

You might even argue why you do this way, if I start with this architecture; that means, at the input it is the same, I have this filter h of n instead. I decomposed this M into some rates M 1 M 2 dot dot dot some MI. This is my y of n, M is basically broken down into capital I number of states. And we know what happens to the frequency response at the output of each of these down samples, we know those effects right. So, this is possibly another architecture, architecture 2, and both of these architectures guarantee us at the output we are still at a rate F naught upon M and this is F naught, this stage it is also F naught.

Now, you might think about a third architecture, which is, I have some filter followed by down sampler, I cascade it with another filter followed by down sampler and so on and so forth. Filtering is doing some kind of operations, where I basically filter out certain frequencies, but the down sampling rate, at the output the net down sampling rate at the output is somehow to be preserved at capital m; that is desired per architecture 1 right.

(Refer Slide Time: 06:18)



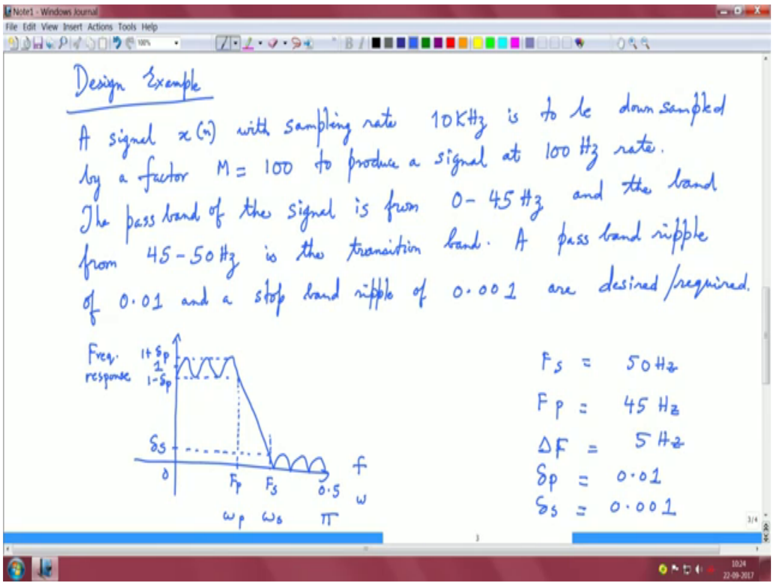
Let us draw the schematic. So, we have x of n. Now this is a little different than 1 and 2, so we have some h 1 n followed by down sampling rate M 1 dot dot dot. Till you know we can go to ice cages, if we want to or we could just restrict it to some say p stages. So, these are cascaded stages and the net sampling rate at this stage is F naught upon M.

So, this is the idea behind multistage implementation, but you may ask this question to start with. It seems obvious that we are greatly increasing the computations by having many intermediate filters at various stages. So, this is opposite to our intention ok. Goal would be, to show why multistage structures are advantages over a single stage, but at stage still you may get a thought in your mind that. If I really have a very narrow transition band right, the filter order that I would require would be humongous. With a single stage let us say you know I am looking at a signal say 10 kilohertz sampled and I want to down sample it by some rate 100 say suppose, and I want a very narrow transition band to accomplish this filtering.

So, the question is, if I use a single stage, we know; obviously, that filter order would be very high, but using multistage designs is it somehow possible to reduce the filter order that is the key thing here, if you realize. You will reduce the filter order somehow using lower order filters at various stages, yet you have to accomplish the specifications of what is required for your design ok.

So, let us start with a design example that will make things very clear right. I mean at this moment it is sketchy we have some thoughts for various architectures, and the thought process is still sketchy for us, because we are not very sure exactly which is the right direction to go about. So, we will see with a two stage design and then we will convince ourselves why this is the right way to go about.

(Refer Slide Time: 11:42)



So, let us consider a signal x of n with sampling rate 10 kilohertz is to be down sampled by a factor M equals 100. So, when we sample down sample this by a factor 100 we get a signal at 100 hertz right. To produce a signal at 100 hertz the pass band of the signal is from zero to 45 hertz and the band from 45 to 50 hertz is the transition band and we are assuming umm equiripple filter. So, we also have some specifications on the ripple in the stop band and the pass band, and that is also part of the design spec, a pass band ripple of point naught 1 and stop band ripple of point naught naught 1 are desired. I mean let me put it, are required; that means, you cannot overshoot this.

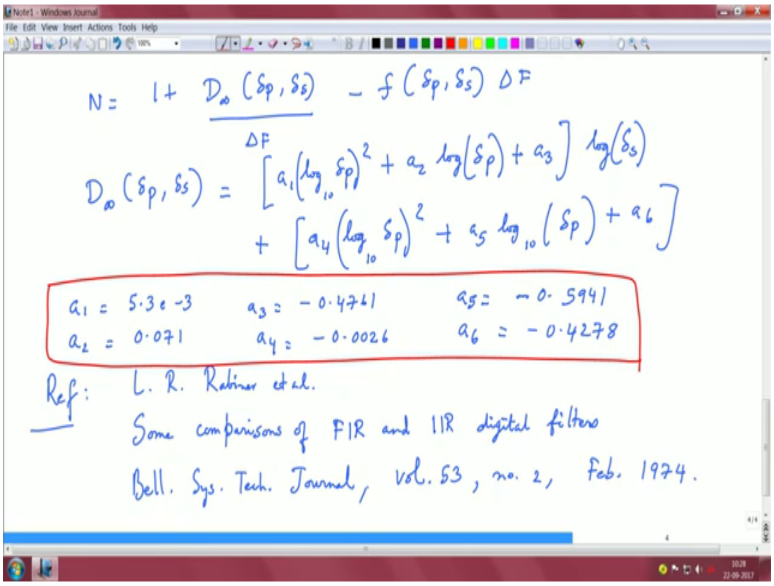
So, if you just think about what this really translates. So, this is our stop band delta s, this is our frequency F or omega. So, this is, this is Fp in the pass band, this is Fs in the stop band and this is 0.5 is normalized frequencies. Writing this is pi at the Nyquist omega s omega p and if this gain is to be 1, this would be 1 minus delta p, this is 1 by 1 plus delta p ok.

So, this is the response the frequency response, this is frequency response on the y axis and in the x axis is your frequencies right you. This is this is this is what we need. So, the first and the foremost step that you will have to take is to estimate the filter order right. So, we are given certain quantities and the if and the quantities are omega s omega p right, the stop band is 50 hertz this pass band cutoff here is about 45 hertz and we need sharp attenuation after 50 50 hertz and we need 100 hertz is, what is the is the is the output rate.

So, we have umm the other parameters; so the delta F which you can compute directly. So, let me say this is, perhaps it is omega, omega is 2 pi F. So, I think I can just put, I can call this as Fs, this is Fp and I call delta F and this is 45 hertz which is 50 minus 45. I mean omega is basically 2 pi F or and then you could normalize this.

Now, we have to get the filter order for this, for this problem and for this we invoke this formula n equals 1 plus d infinity of delta p comma delta s and I think I have to mention the other two parameters as well here, delta p is point naught 1 and delta s is point naught naught 1 ok. So, now, we have all these specifications here.

(Refer Slide Time: 19:08)

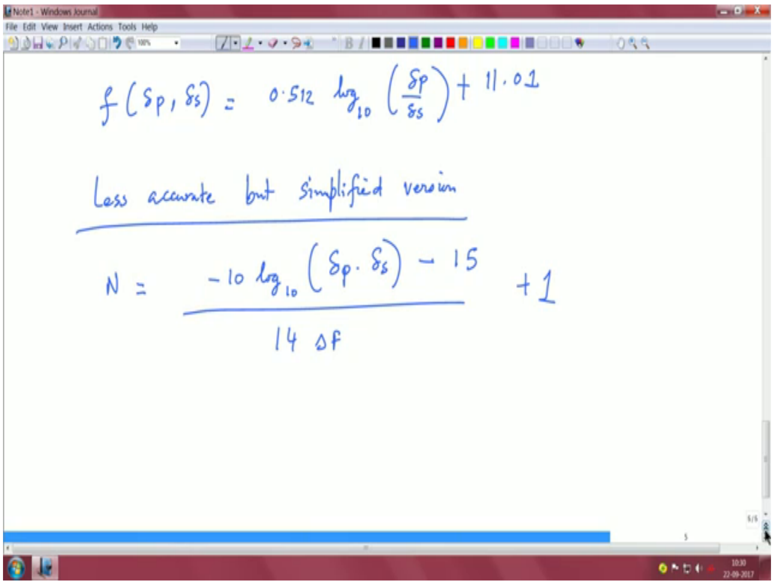


Now, from this we have to get towards estimating the filter order. So, we have an equation towards this and I will provide you the formula and I will give you the reference. So, first let me write the reference. So, this comes from the work of Larry Rabiner from Bell Labs umm from Bell Labs in the 1970s and this paper is basically titled some comparisons of FIR and IAR digital filters. This is from bell system technical journal volume 53 number 2 February the issue 2 and february 1974 and let me give you what these formulas look like.

So, D infinity of delta p comma delta s is a 1 log delta p to base 10 square plus a 2 log delta p plus a 3 times log delta s plus a 4 log delta p to base 10 square plus a 5 log to base 10 delta p plus a 6 and the parameters are a 1 is 5.3 e minus 3 a 2 is point naught 71 a 3 is minus 0.4761 a 4 is minus point naught naught 26 a 5 is minus 0.5941 a sis is minus 0.4278, it looks quite messy formula is empirical.

And there is also this term and we can ignore this to some extent, but I will write, I will write down the formula as well for what F is, will have to go to the next page, this is not enough F of delta p comma delta s is 0.512 log to base 10 delta p over delta s plus 11.01. So, if you are interested in figuring out how they arrived at this formula, you can look into this paper which has some other reference papers and that will give you some insight, how this is done this is an empirical formula.

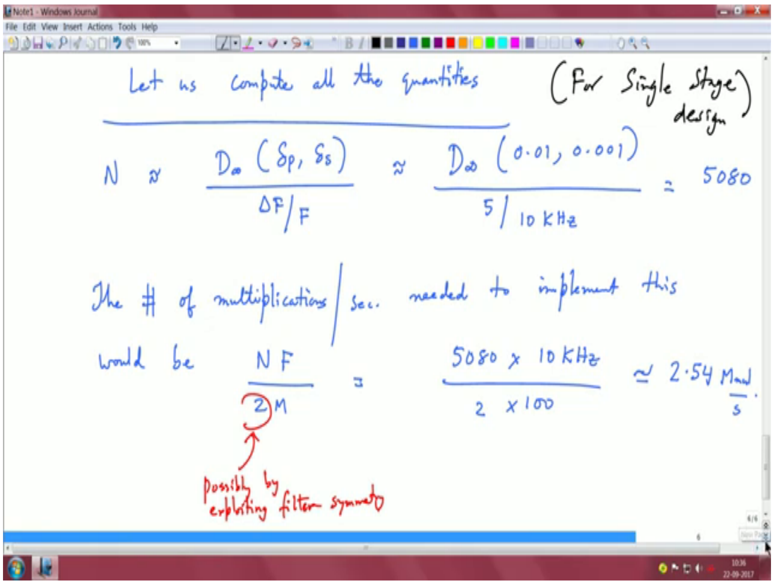
(Refer Slide Time: 23:41)



And there are some simplifications which are less accurate I would mention this as less accurate, but simplified portion and that formula is n equal to minus 10 log to base 10 delta p times delta s minus 15 by 14 delta F plus 1. So, these details are given in Rabiners work, but for all purposes for this design example, we can ignore this test or we can ignore F this, this function; that is this, this other approximation that we have, will ignore this term and we will focus only on this ok, and you have all the numerix here, you can, you have delta p, you have delta s, you have all the values, all the constants that are connecting.

So, you can plug in to the formula and if you are not comfortable with this formula you could also use Belangers formula which we give, which, which I gave you earlier; that is also a formula that you can use right, you can use whichever is your favorite. A small thing that I noticed here is, it has to be delta F by F on this slide and in the previous one as well, where we have, this has to be delta F by F and delta F upon F ok. So, that the units come out correct.

(Refer Slide Time: 25:42)



So, let us get towards computing these quantities. So, let us compute all the quantities from the specifications that we have ok. Now umm N is approximately the infinity of delta p gamma delta s we ignore the minus 1 factor there right, and we will say this is going to be delta F by f. So, if you plug in the delta p and delta s that we have, which is point naught 1 point naught naught 1 divided by delta F is 5 hertz, F is basically 10 kilo hertz.

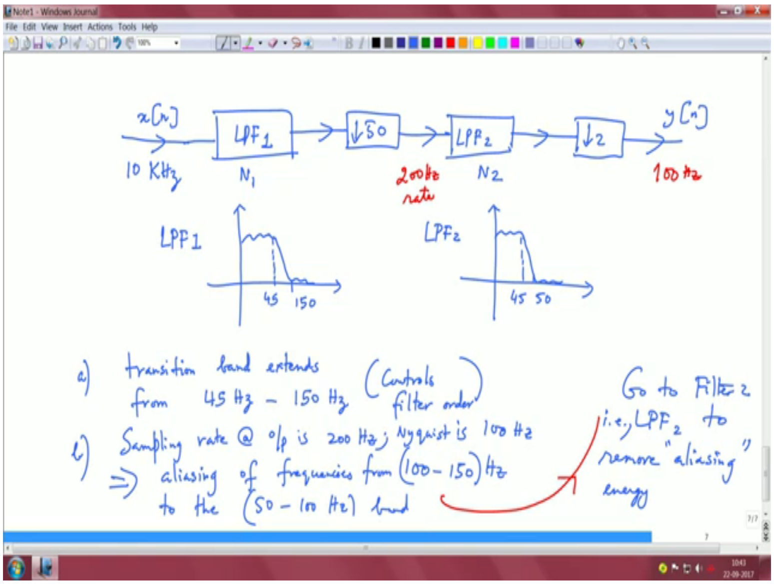
So, you see the change here 5 hertz is the transition band, 10 kilohertz is your input input rate and that is why things get skewed so much umm. So, you do the maths and the answer to this would be 5080, it is a tremendous filter order 5000 is the order of the filter that you would require. This is a practical problem right, I had a 10 kilohertz signal and I wanted to down sample it by 100 and have very narrow transition band of 5 hertz and if I did all the maths, this is what I will land up with.

So, now you can imagine the latency is etcetera that this filter will cause if you were to implement them. Now let us look at the multiplications required per second needed to implement this. The number of multiplications per second needed to implement, this would be N times F over 2 M, I mean I am bringing this factor 2, I think you can question why possibly by umm exploiting filter symmetry or other properties. So, assume you can do something clever. So, I am bringing this factor 2 and that factor 2 would hold good for any multistage filter. So, that is something that you have to bear in mind, but if you are not comfortable putting this factor 2 is still right.

If you did the blunt way without any optimization or any umm special tricks on the FIR filter, you would just have a factor N F upon M ok; so just a, just to make you aware of the factor 2 that we, why we need. So, if you just plug in these values the filter order be computed to be 5080 times, sampling rate 10 kilohertz divided by down sampling rate is 100 justice factor 2, that would appear and if you did this, it roughly comes to about 2.54 mega multiplications per second. This is quite a lot.

You are burning and you just compute with your with your circuit elements, what it would take to burn this much power per second, it is a lot. So, you are wasting power, because your multiplications are huge. Second you have to consume a lot of area, if you were to put this in to be a silicon right, because the filter order is huge and you have to basically have all those taps stored, and then what about leakage and many other factors that you have to bring it right. So, this is not a good solution ok. So, this is for single stage design and this is not very efficient ok.

(Refer Slide Time: 31:03)



So, let us see if we can do this slightly differently. Now from the intuition we have from architecture 3 right. So, architectural 2 is still no good, because you have the base filter which is 5000 length long and then if you just filter it with that, still you are going to burn that, you know the that much amount of power, because you are filtering through a filter which is a huge filter order right. So, we still think architecture to is going to be no good, except that you could perhaps simplify certain things spectrally by using a cascade of these down samplers, but let us directly plunge into architecture 3 and see how we can get any benefit.

So, we have x of n, which is at rate 10 kilohertz. So, we first subject this to a low pass filter 1. Let us say this order is N 1. Now I can write down sampling by 100 as down sampling by 50 and down sampling by 2, I could do that.

So, the first stage, let me down sample this by 50, then I take this to another low pass filter 2, then I down sample this by a rate 2 and then I get y of n and let us say this order is N 2 ok. So, let us look at some interim points. So, if you have 10 kilo hertz and your down sampling by 50, at this step you are at 200 hertz rate right and from 200 hertz, if you down sample by 2 you are at 100 hertz rate which is, what is what is the, what is umm desired right. So, now, from 10 kilo hertz you are at 200 from 200 you are at 100.

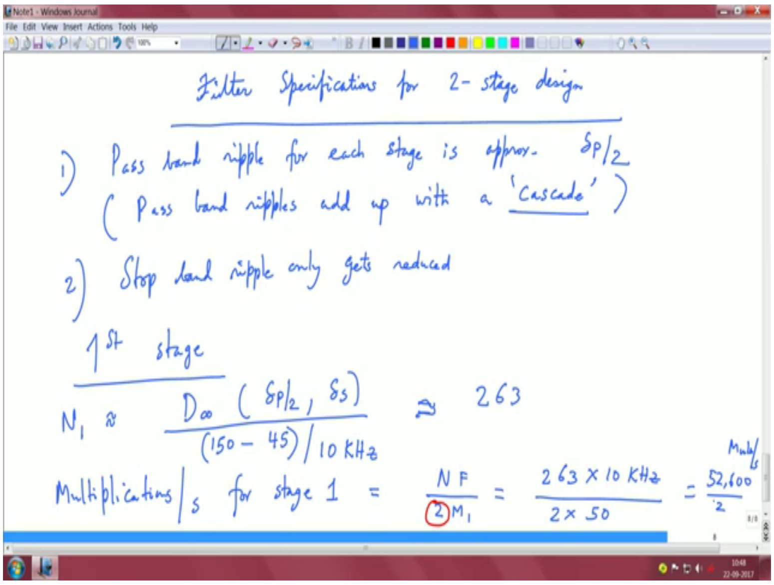
Now, what we require is some specifications to the filters right; that is the missing piece, some specifications for the filters, because I have not given you the specifications of your individual multistage filters. I gave you an overall specification of what the desired frequency response has to be right. So, one can think about the following here, for this stage umm. I am keeping it flat, but you know we just imagine in your mind sort of somebody pulls here. Let us assume that this is 45 to 150 right. I am stretching my transition band right. Instead of having 45 to 50 the original I am making it 45 to 150 and this is as usual 45 is probably sharper 45 to 50.

So, this is some careful detail that you will have to note. So, here this is for LP F 1, this is for LP F 2 transition band extends from 45 hertz to 150 hertz and is advantageous to us, because if we stretch the transition band the lower the filter order right. So, this makes sense, but there is another issue that you have to worry about is the aliasing, because if you had this rate at 200 hertz right that is your, your rate is for 200 hertz, your Nyquist would be at 100. So, the Nyquist is at 100 hertz any frequent. All those frequencies from 100 to 150 hertz would be aliased back between 50 to 100 hertz here and our intuition should tell us that if we eliminate 50 to 100 hertz using another low pass filter here, we are removing the aliased components clear.

So, I mean while drawing this response 2, things have to be borne in mind; one is the transition band, because this controls the filter order, but sampling rate at output is 200 hertz which implies aliasing of frequencies from 100 to 150 hertz to the 50 to 100 hertz and which sort of motivates us to go to filter 2 which is this low pass filter to remove aliasing energy or aliased components right.

These energy aliased here and we want to remove the aliased components and then that sort of gives us the idea why we need the second low pass filter to be within 45 to 50. This is a transition band 45 to 50 hertz is the transition band.

(Refer Slide Time: 38:41)

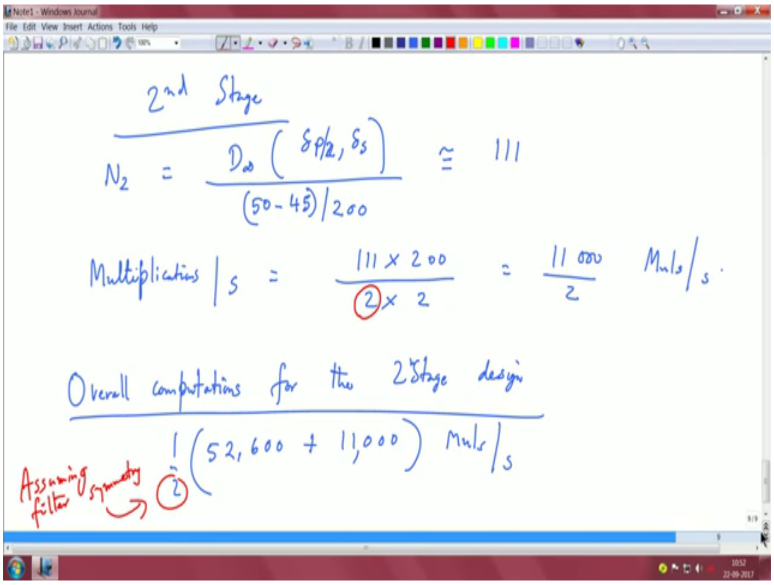


So, let us, let us do umm our computations for this umm this design. Now there is a little bit of detail on the ripples, we saw the frequency domain effects both in terms of aliasing as well as in the transition band and we have to also see what happens to the ripples. So, this is from observations pass band ripple for each stage, is approximately delta p upon 2, because when you cascade the pass band ripples add up and stop band ripple only gets reduced. So, therefore, we are to leave our stop band to delta s, but the pass band, we have to consider it to be delta p upon 2 for individual filters.

Now, let us look at the first stage. So, for the first stage N 1 is approximately D infinity of delta p upon 2 comma delta s divided by. Now the transition band delta F is now 150 minus 45 and your frequency at the input is 10 kilo hertz and if you read the math, this would give us 263. So, where was 5080 and where we are at 263 right, and you could do your multiplications per second for stage 1 is equal to N F upon. Again we will assume some factor 2 here, assuming that the filter is done efficiently, I will just mark it with a circle. If you if you are not comfortable you do not have to use this too, it is perfectly fine.

So, this would give us 263 times 10 kilo hertz upon 2 times 50 right and whatever that that number is now. So, I will just write this as from what I have is 52 600 by 2 multiplications per second justice factor 2 appears the way it is.

(Refer Slide Time: 43:05)



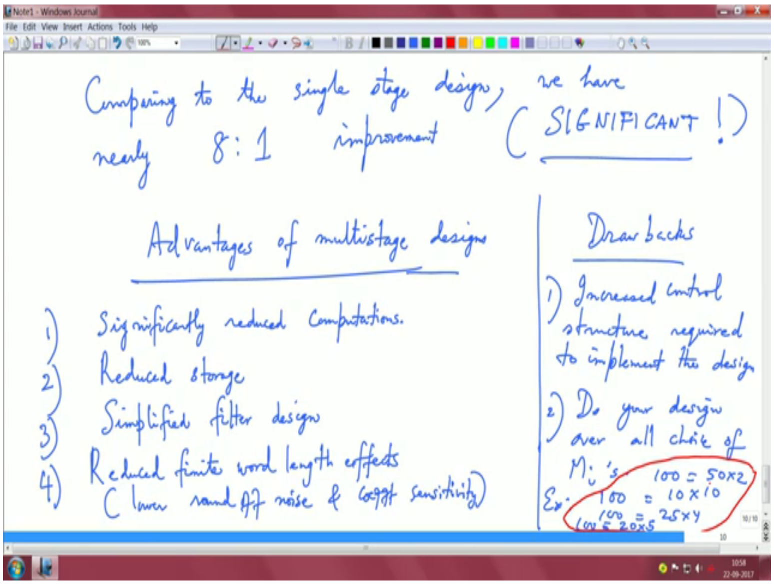
Now, let us look at the second stage. For the second stage we do the same way and 2 is the infinity of delta p delta p comma delta s. So, umm for the delta p upon 2 comma delta s and the transition band is between 50 and 45 hertz divided by the input rate is now 200 hertz right. We had 10 kilohertz, we down sampled it by a rate of 50 and the rate was 200. I mean we were at 200 hertz and that is down sampled by 2; that is and we have you know 100.

So, therefore, this is 200 here, because this is the input for the second stage and you do the math for this. This is approximately 111 umm taps, means fil this, this is the order of the filter and you do the multiplications per second, this is 111 times 200 divided by 2 times 2, 2 is your M, M 2 here right. So, this umm I will leave it as 11000 divided by 2, this 2 factor appears all the time, I just would put this in this mark this by red. If you are not comfortable you do not have to use this factor. This is the number of multiplications per second.

So, now the overall computations for the 2 stage design is.We said it was 52600, I guess 52600 plus 11000 one half. So, many multiplications second and this is assuming some filter symmetry that we want to bring in to simplify this design.

So, I think if we just compare what we have here versus the single stage design, which had this 2.54 mega multiplications versus something that we have in the order of about 60000 by 2 which is 30000 multiplications right, it is a factor of 1 is to 8.

(Refer Slide Time: 47:01)



So, this is something which we have to bear in mind. So, comparing to the single stage design, we have nearly 8 is to 1 improvement and this is really significant. This is like 8 x of improvement that we have with 2 stage design and one can imagine that in the early stages, the sampling rates are high and equivalent equivalently the transition widths are also large, leading to smaller values of the filter length and in the last stage, the sampling rate and transition bits tend to be smaller giving you know lower orders, but possibly comparable to the earlier stages.

So, this is something which you have to, you have to note. So, let us list some advantages and disadvantages of multistage architectures. So, first is significantly reduced computations and we saw this through the example, then the filter taps are also less right, because the order is less. So, therefore, it is reduced storage, then our filter design is also simplified and I think one of the most important things that people often ignore is, you know you never have floating point when you think about implementation, it is fixed point.

So, therefore, when you think about such filters they are subject to finite word length effects, and if you think about the quantization noise, you have to look at what is the implication of quantize, when you, when you quantize every filter and you take an input noise and you filter it through this filter. I mean there it is going to be colored, because of this filtering end and you know the variance also would be higher.

So, therefore, it is important to have reduced filter orders, so that we have reduced finite word length effects and therefore, lower round of noise and sensitivity issues. This is very important and particularly if you are launching a satellite, putting these filters etcetera you would be very very careful these about quantization effects; so reduced finite word length effects, which means lower round of noise and coefficient sensitivity.

Now, if we were to even think of drawbacks, I would not say really there are any drawbacks, it is a win situation, but I think it is, but I you know we just have a list, you know pondering about it. So, one is increased control structure. Increased control structure required to implement the design. I mean you have to be slightly clever how to, how you stitch the blocks together. There will be timing issues and many other things that you have to take care. It is a little bit of implementation detail and I think then you have to really do your design over all choice of a mice

For example if I want to realize 100, I might do it 50 times 2, I may do umm example 100 could be realized as umm, I just put this in this form 100 is 10 times 10 125 times 400 is 20 times 5 and we also looked at the case where 100 is 50 times 2. I mean there are all possible combinations of if you want to think about a 2 stage design, whether you want down stay down sample by 50 first or 10 or 25 or 20 and then cascade it with either 2 10 4 or 5 right. You have all these choices that you will have to think and ponder upon. You have to do this, I mean I think that is I do not see it is a drawback; it is just a burden on the designer to be a little more efficient and clever to exercise this control.

So, I think this completes this lecture on multistage implementations for realizing umm high rate decimation filters and hopefully you find this topic useful, when we actually build circuits and design you know and do design architectures for multirate systems with multi stage designs.