Digital Voice & Picture Communication

Prof. S. Sengupta

Department of Electronics and Communication Engineering

Indian Institute of Technology, Kharagpur

Lecture - 31

MPEG-1 Audio Coding

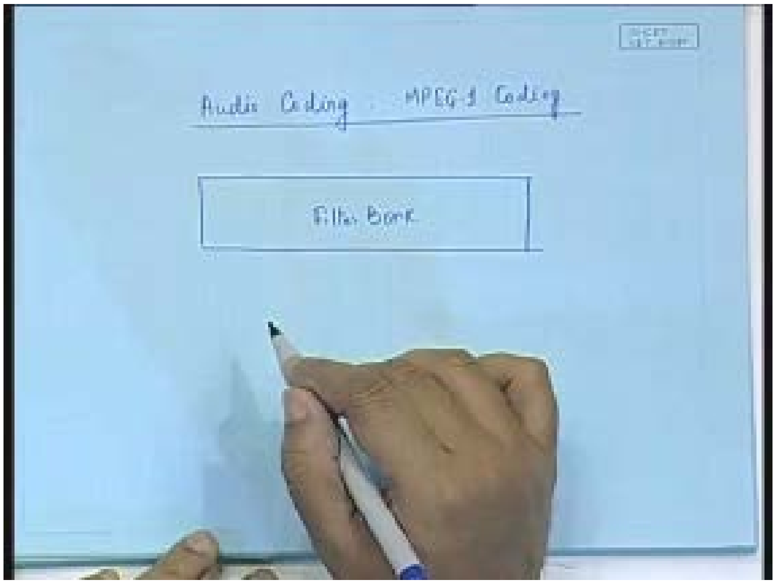
..............about the, I mean, in the discussion of audio coding we will be paying our attention on the MPEG-1 codec and especially we will cover the polyphase filter bank implementation. So this is MPEG-1 coding part, this is what we are going to discuss.

In the last class when we were talking about the AC-3 decoders and then we happen to talk about the analysis and synthesis filters for the multichannel or multiband filtering that is what we were talking and there we had seen that there are two basic aspects towards the implementation of such multiband filters. One of the ways is that we implement the FIR filter in the time domain but there we happen to add their frequency domain characteristics so some kind of an overlap and add techniques we have to adopt in the frequency domain and we apply a frequency domain alias cancellation that is one of the approaches. And the second approach is to first window the samples and after that we are going to talk about the....... and that is followed by the time domain alias cancellation approach whereby we take the transforms of the windowed samples and then we consider the filtering in the transform domain. And we had seen that this is what we apply in the AC-3 audio codec and the same thing is also applied to the MPEG-1 audio coding also.

Just excuse me for a minute............

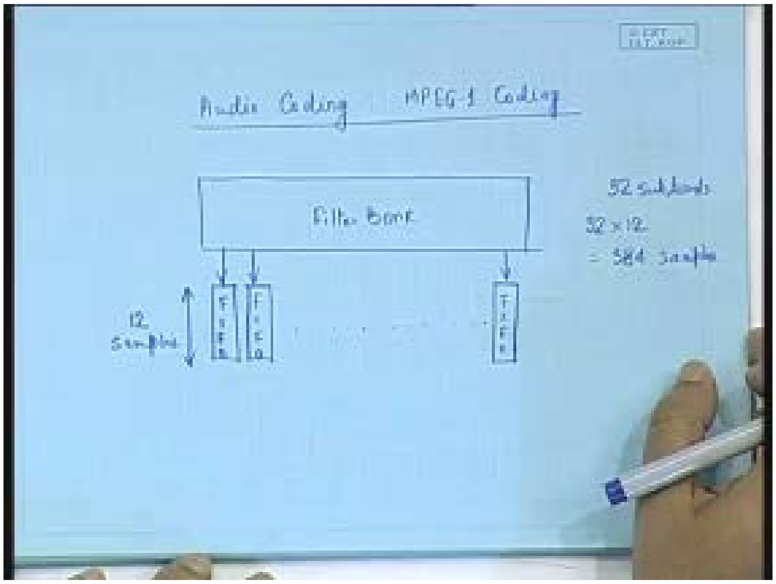
We will first start with the MPEG-1 audio coding basics. There what we do is that the basic flow diagram that is followed in the MPEG-1 audio coding goes like this. We have a filter bank first and this filter is realized in the form of a multiband filtering and in multiband filtering we apply the same technique that means to say the time domain alias cancellation and we adopt a frequency domain filtering rather than the time domain filtering. So this filter bank implementation we will be discussing very shortly.

(Refer Slide Time: 4:36)



Once it is filtered; actually it is filtered into 32 different subbands. This 32 subbands would corresponds to 32 different critical bands based on the perceptual audio and then we will be having for each of this 32 filter banks we keep one first in first out kind of buffer. So these are the FIFO buffers and we have FIFO buffers for each one of the samples for each of the subbands. So there are 32 such FIFOs and each FIFO is 12 samples deep, so these are 12 samples so this can store up to 12 samples so that because there are 32 such subbands and we have 12 samples per subband so we have totally 32 into 12 that means to say 384 samples they form one audio frame in MPEG-1.

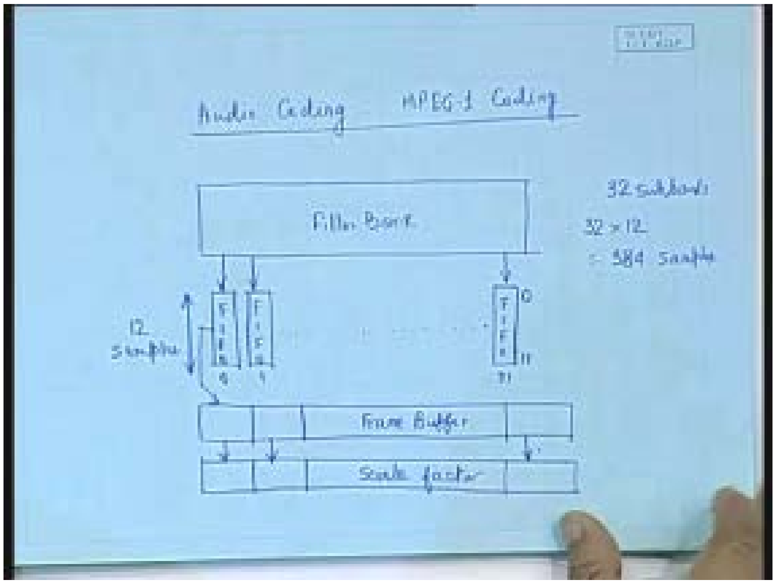
(Refer Slide Time: 5:51)



Now all these FIFO outputs, these 12 samples together they are accumulated in the form of a buffer. So this buffer the frame buffer will have a total capacity of 384 samples so we have 12 from the subband number 0, 12 from subband number 1 and etc up to 12 from subband number 32. So this is FIFO number 31, this is FIFO number 0, this is FIFO number 1 and it is storing 12 samples. Now this frame buffer this contains the 12 subband data, I mean 12 points for each of the subbands, 12 samples for each of the subbands. Now what we can do is to find out that amounts this data whichever is having the highest value based on that we need to do some scaling on the data so as to have the maximum usage of the dynamic range of the samples.

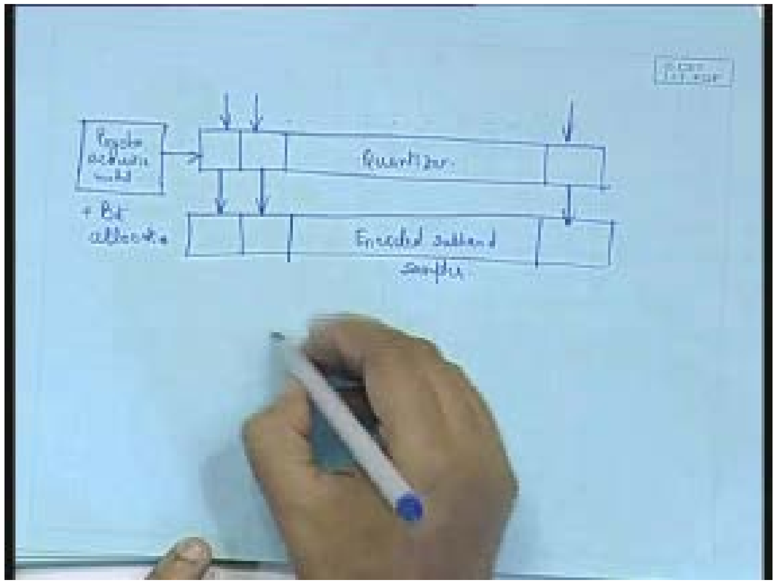
So the frame buffer is put through a scale factor so there will be different scale factors for different subbands. So we have a scale factor buffer which will store the scale factor values or where the scaling factors will be decided based on the sample values what we have in the frame buffer based on the maximum value of the sample and the scaled values will be stored in a buffer where the quantization will be applied. So the quantization will be applied on this scaled value of samples.

(Refer Slide Time: 7:55)



So what we have is that from this scale factor buffer this 32 subband samples whatever we are getting this will get quantized. So this is for the subband 0, subband 1 up to subband 32 so there will be quantizer; so the quantizer will be chosen based on what; based on the psychoacoustic consideration. So we have a psychoacoustic model and bit allocator. So psychoacoustic model, as you know that this works on the masking characteristic. The psychoacoustic model plus bit allocation and the scaled sampled values are quantized through this quantizer and then the quantized samples are put into the put for encoding. So here we obtain the encoded subbands samples for all the 32 subbands.

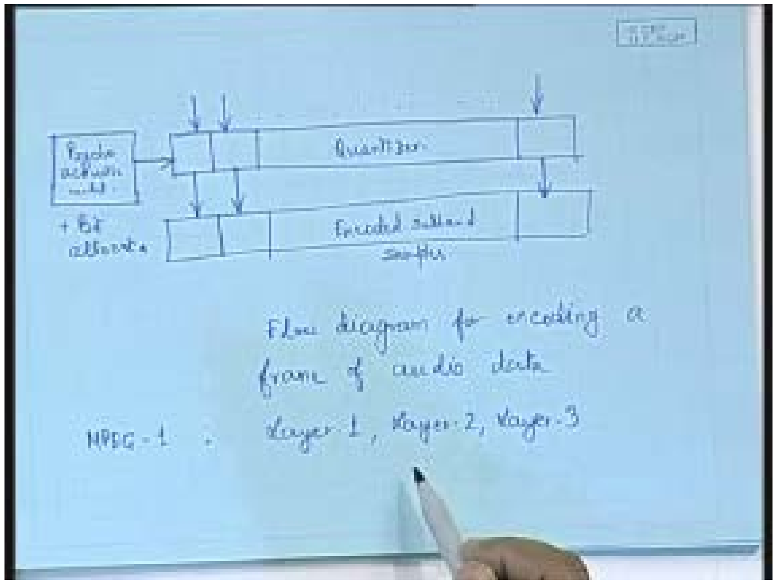
(Refer Slide Time: 9:26)



This is the flow diagram. I could not draw in one page this was the first page so we start with the filter bank then the frame buffer then the scaling is applied and after scaling, the samples are quantized, the quantization is decided by the psychoacoustic model and bit allocation and then we have the encoded subband samples. So this the flow diagram for encoding a frame of audio data.

In fact this form of encoding is the simplest form of encoding. In fact MPEG-1 standard that specifies three layers of encoding and these are the layer 1 encoding which is the simplest then followed by layer 2 and the last one is the layer 3 which is the most complicated one; although in terms of the compression efficiency layer three will be the best but for applications where we do not require that much of compression efficiency, there layer 1 implementation happens to be the simplest. In fact there are two psychoacoustic models which are defined: the simplest is the psychoacoustic model one which................ I mean, the layer 1 and the layer 2 is based on the psychoacoustic model one whereas the layer 3 uses psychoacoustic model 2 which is a more complicated psychoacoustic model.

(Refer Slide Time: 11:20)



In layer 1 we find that a frame is composed of 384 samples whereas in the case of layer 2 a frame is composed of 1152 samples. So there what we have is that this FIFO buffers in the case of layer 1 is only 12 samples deep and there we have 36 samples for every subband. So 36 subband means that we have 36 into 32 which mean to say 1152 total samples we have and this 1152 samples, this 36 samples per subband they are again divided into 3 blocks of 12 samples each. So each block will function like a layer one encoding and in fact the scaling factor can be applied differently for each one of these blocks. So, for each 12 samples we can apply different scaling factors.

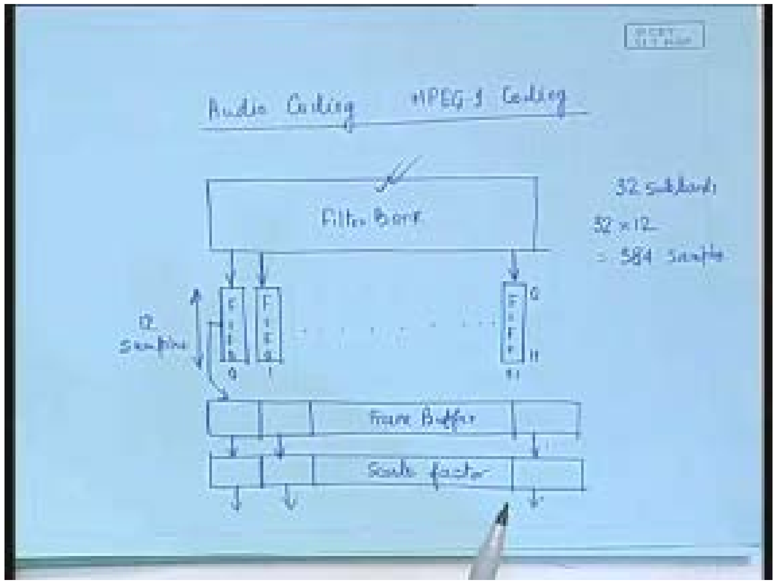
Now one thing which you can see is that the scaling factors need not be the same for all the subbands. We are not applying uniform scaling factors to all the subbands; rather the scaling factors are different for different subbands.

Now we come to the implementation of this filter bank. As I have already told you that instead of going in for an FIR filtering approach in the time domain and then doing the alias cancellation in frequency, we adopt a windowing followed by the DCT which in this case happens to be the modified discrete cosine transform the MDCT and then we have the alias cancellation in the time domain by adopting the overlap and add technique which we had already discussed in the previous class. Let us come to the filter bank implementation.

Yes, any questions? [Conversation between Student and Professor – Not audible ((00:13:222 min))] these are already coming. See, these filters are already the samples that we obtained through the subband filters. So they are already the filter bank outputs so they have been already MDCTed. Actually in MPEG audio coding standard one thing which..........I think that I mentioned a couple of classes back where I was showing the generalized block diagram of perceptual audio coding where I had mentioned that the PCM samples that we obtain that goes through two different paths: one path is where we compute the FFTs and the idea of calculating the FFTs there was to obtain the tonal components based on which the psychoacoustic model acts and there the bit allocation is decided. The other part is the direct encoding path.

Now we are talking in this case about the direct encoding path and in order to encode what we do is that we adopt this kind of a filtering approach whereby we are applying the windowing followed by the MDCT and the reversal will be done at the decoder. Now we are talking only up to the encoding part over here. Now we will be talking about the filter bank implementation; that how we implement the filter bank.

(Refer Slide Time: 15:04)



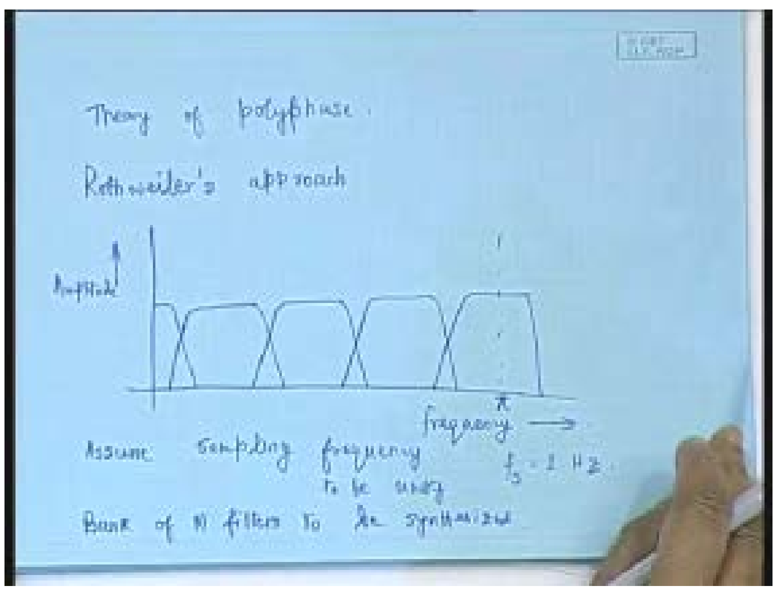
Now, before we do that let us go into some theory behind the polyphase filter. So we will talk about the theory of polyphase filters. Why we call it polyphase; see, it is because........... so basically this approach is first proposed by Rothweiler and in Rothweiler's approach a multi-phase filter implementation has been analytically established. It is shown that if you are having a prototype low pass filter then you can realize a bank of filter by simply modulating that filter; what you have to do is to multiply the filter transfer function.

Or, if you are considering the filter in the z domain what you have to do is to modulate that by the center frequency of the band pass filter and if you have a bank of band pass filter to be implemented then you can simply multiply by that band pass filter center frequency and you can realize a bank of filter like that. This was Rothweiler's approach and this approach we will be making use of and we will be showing that how this approach basically translates to the same way of implementing a windowing function followed by the DCT, so we will show that.

What we have to do is to have a bank of band pass filter. Let us say that we are showing a bank of band pass filters whose responses are like this (Refer Slide Time: 17:13). So here we have the amplitude and here we have the frequency. Now what we do is that we first start with a low pass filter. So it is a low pass filter which is followed by a band pass filter and then we will be having bank of band pass filter that goes like this.

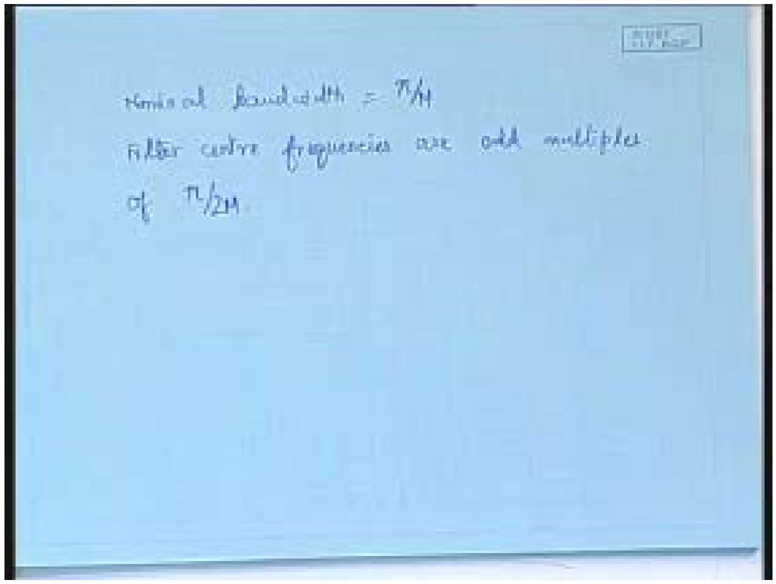
So we assume so we assume that the sampling frequency is unity. So assume the sampling frequency or rather f s you can write; f s is assumed to be 1 Hz so that the frequency to be covered is 0 to......... if it is 1 Hz then what is the maximum angular frequency that you are talking of; that is that is to be 2pi divided by 2 that means to say pi; pi will be the maximum frequency. In fact 2pi radians per second will be your sampling frequency so that the maximum frequency that you are talking of is pi and you are assuming that there are bank of M filters to be synthesized, so bank of M filters to be synthesized.

(Refer Slide Time: 19:21)



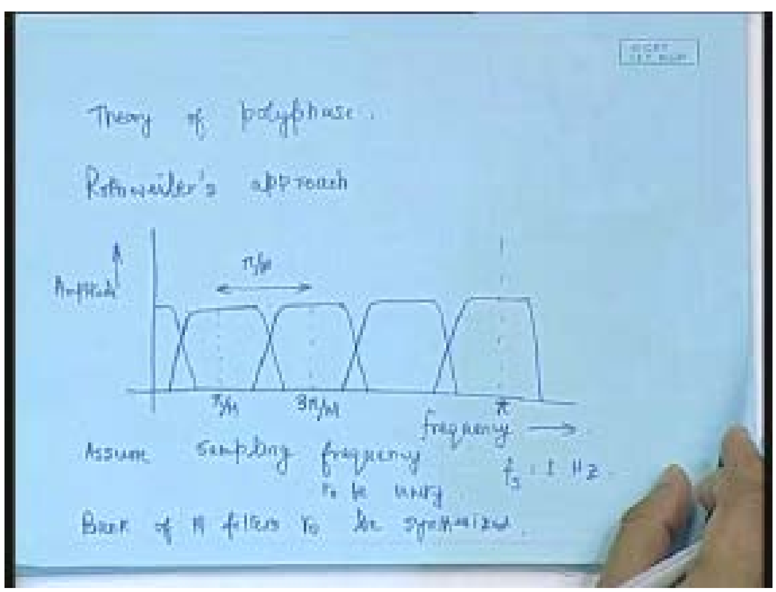
So what is the nominal bandwidth for each of the filter?
The nominal bandwidth for each of the filter is going to be pi by M. So the nominal bandwidth is going to be pi by M and the filter center frequencies are odd multiples of 2pi by M pi by 2M.

(Refer Slide Time: 00:20:17)



In that case this center frequency what we are talking of (Refer Slide Time: 20:25) this is going to be pi by M then this is going to be 3pi by M and this is pi and the bandwidth is going to be pi by M, so here to here we have pi by M or if you are taking from here to here that is pi by M.

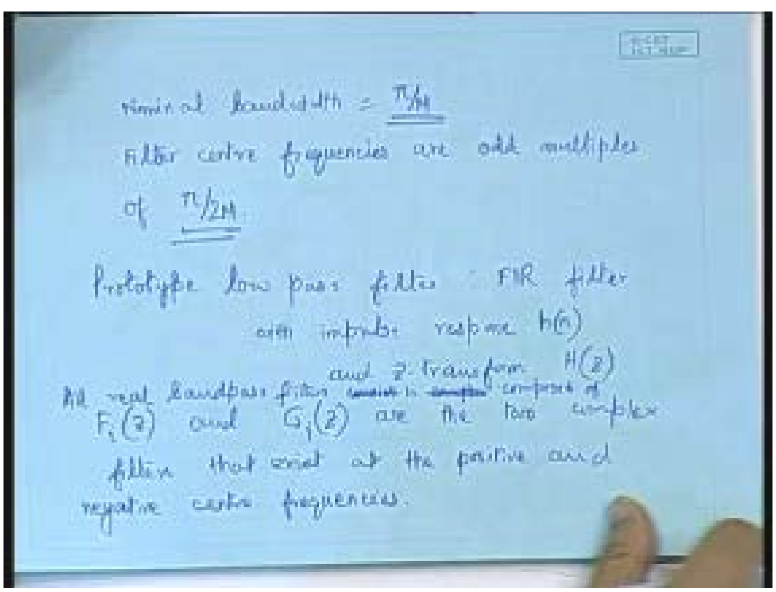
(Refer Slide Time: 20:53)



[Conversation between Student and Professor – Not audible ((00:20:15 min))] pi by 2M, yes, pi by 2M and 3pi by 2M sorry yes correct; center frequency is pi by 2M and 3pi by 2M as we have already written down that the nominal bandwidth is pi by M and these are odd multiples of pi by 2M so that we have pi by 2M, 3pi by 2M, 5pi by 2M and so on. Now, the band pass filters the prototype low pass filter we assume that it is an FIR filter with impulse response h of n. So the prototype low pass filter we assume that it is an FIR filter with impulse response h(n) and its z transform is H of z.

Now the band pass filters, they are composed of two complex filters which are F i(z) and G i(z) these are the two complex filters that exists at where; at the positive and the negative frequencies. So all real band pass filters, so we should say that all real band pass filters............. so this is composed of it is composed of two complex filters is sorry composed of is composed of two complex filters F i(z) and G i(z); F i(z)'s center frequencies are all located in the positive frequencies and G i(z)'s center frequencies are all located at the negative frequencies. So, the two complex filters exist at the positive and the negative center frequencies.

(Refer Slide Time: 24:06)

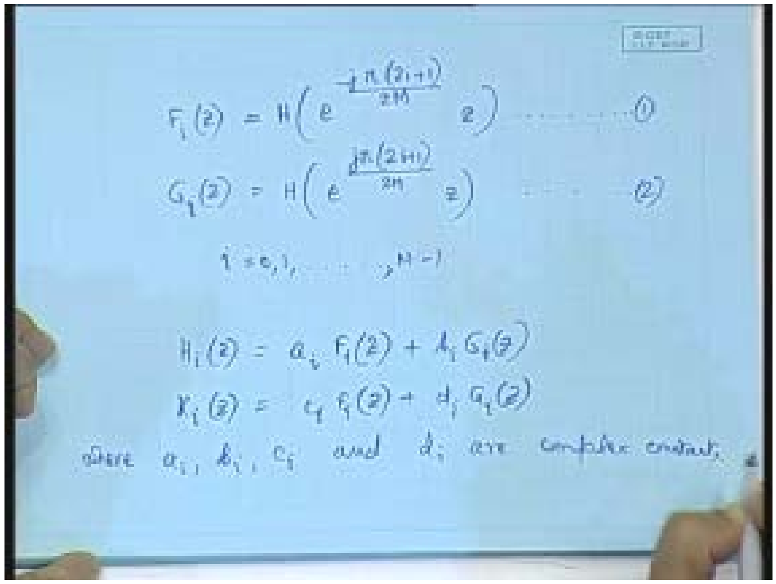


So what we have to do is that we can express the F i(z) and G i(z) as follows. So F i(z).......... in fact the suffix i stands for the ith filter, the ith band pass filter we are talking of; so F i(z) will be given as H, H of z being the transfer function of the......... is the z domain transfer function of the low pass prototype. But in this case F i of z will be not H of z but it will be the complex exponential involving the centre frequency so it will be because the centre frequencies are all odd multiples of pi by 2M that is why F i will be e to the power minus j pi into 2 (i plus 1) divided by 2M so this into z.

So if you are substituting i to be equal to 0, in that case what you get, you get (2i plus 1) to be equal to 1 so it is minus j pi by 2M so this is the first center frequency. So the first center frequency is located at pi by 2M. So f of 0 you obtain by simply multiplying by e to the power minus j pi by 2M z so that realizes the F 0(z) component and the G 0(z) component or in general the G i(z) component will be written as H of e to the power j 2pi and not the minus sign because this refers to the negative center frequencies that is why it will be written as e to the power j 2pi (2 i plus 1) by 2M z so that together this F i(z) and G i(z) that realizes the real filter.

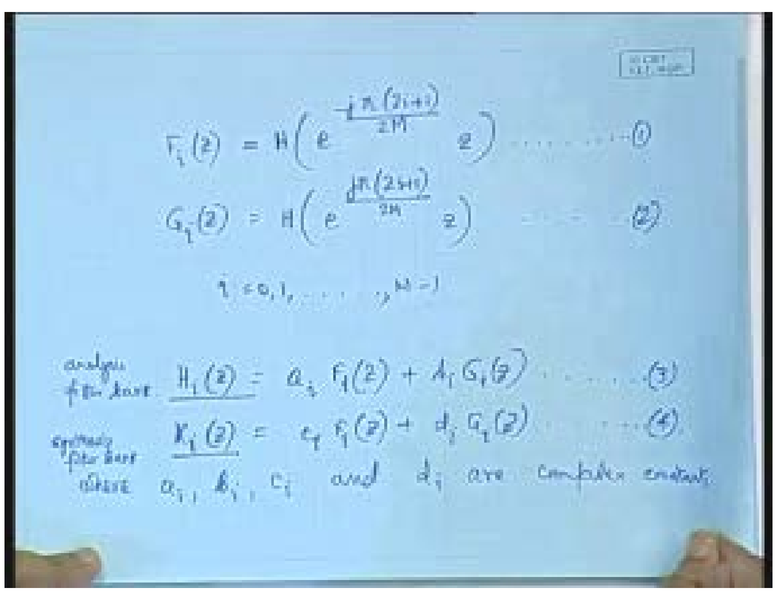
We can number this equation. Let us say that this is equation number 1 for us (Refer Slide Time: 26:30) and this is equation number 2 so equation number 1 and 2 what we have derived is the complex filters in z domain, what we have obtained by multiplying the H of z multiplying z by the exponential of this e to the power minus j pi (2 i plus 1) by M and we get a band of filters; using 1 and 2 we get a band of filters for i is equal to 0, 1,....... up to M minus 1 so we get M such different filters. And we can in fact obtain the overall filter response H i of z as what; we can have some constants, we use constants a i b i c i and d i whereby we can express this overall transfer function H i of z as a i F i(z) plus b i G i(z) and likewise we can apply the we can have...... so this is the analysis transfer function H i of z the combined analysis function and then we can also express the corresponding synthesis filter K i(z) as c i F i(z) plus d i G i(z) where a i b i c i and d i are complex constants.

(Refer Slide Time: 28:39)



Now, H i of z forms the analysis filter bank for i is equal to 0, 1,..... up to M minus 1 and K i(z) forms the synthesis filter bank. So this is the synthesis filter bank and this is the analysis filter bank (Refer Slide Time: 29:07). So, pictorially we can show like this: this realization let us say that this is equation number 3 and this is equation number 4.

(Refer Slide Time: 29:20)

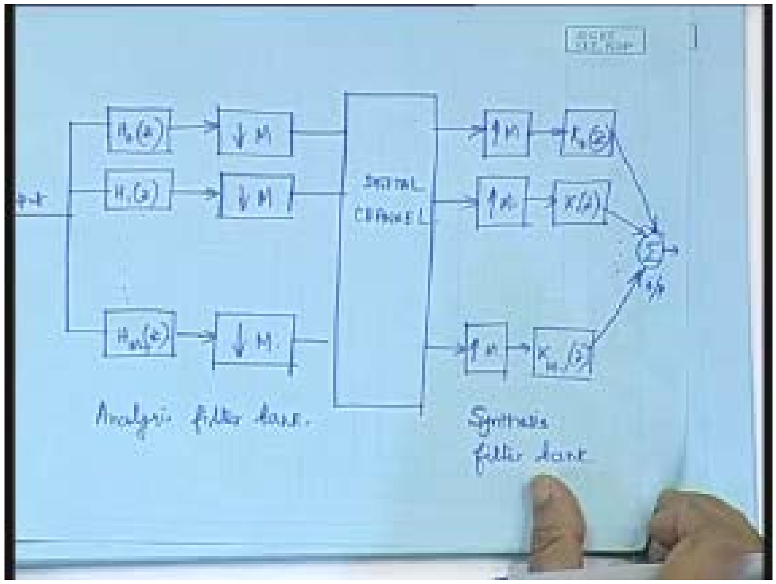


Pictorially we can show like this that this is the input, it is put through M different filters consisting of M different band pass filters consisting of H 0(z), H 1(z),......... up to H m(z) and because each of these bands happen to have 1 upon M times the original bandwidth there are M different filters so the total frequency range is pi and each of them as I have already shown you that each of them covers only pi by M as the bandwidth.

So each filter bank has a bandwidth of pi by M so we need not have to use the sampling frequency the original sampling frequency rather we have to decimate and decimate the sampling frequency by a factor of M. So there will be decimation by a factor of M down sampling, so down sampling by M that we have to do for all these filter banks and then it goes into the digital channel. So this is the encoder part or rather the analysis part; so this is the analysis filter bank and correspondingly in the synthesis filter bank or the synthesis realization what we have to do is just the reversal; so we have up sampling by M and that we have to do for all these filter banks; there should be an up sampler by a factor of M and this should be followed by followed by the K 0 the corresponding synthesis filter bank.

So here we should have a filter K 0(z) here, K 1(z) and the last one will be........ oh this this should be M minus 1; you did not point out this mistake to me so this is H m minus 1 and this is K m minus 1(z) (Refer Slide Time: 32:14). Now what we have to do in order to compute the output; in order to obtain the output we have to combine the responses of all these synthesis filters. So there should be a summer where we are going to combine the outputs of all these synthesis filters and here we will be obtaining the output so the output will be obtained over here; so this is the output of the filter. This is the synthesis filter bank. This output should be exactly equal to the input if it is a perfect reconstruction. If the H 0 and the z 0 are so chosen that the or rather the H i’s and the K i’s are so chosen that the output becomes equal to input, in that case that is the condition for perfect reconstruction.

(Refer Slide Time: 33:30)

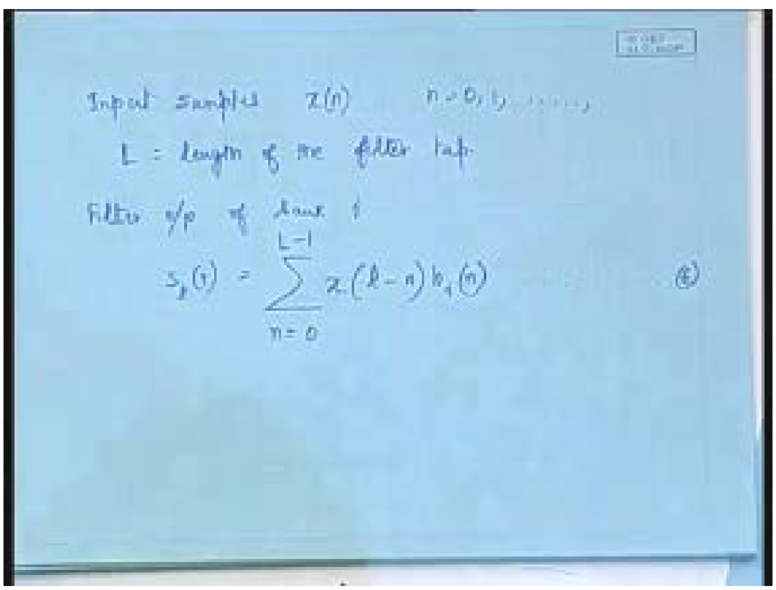


Now, if in this equation what we have written as the linear combination H i(z) the analysis filter and the synthesis filter written as the linear combination of this F i(z) and G i(z) if there we impose a condition that a i is the complex conjugate of b I; a i b i c i d i all are complex constants I already mentioned; so if I choose a i to be equal to b i star in that case the impulse response of H i what we obtained; the impulse response of H i this happens to be real and it is given by................ (Refer Slide Time: 34:27) if this is so the impulse response of H i is real and is given by h i of n equal to the real part of a i into cosine; this comes through a simple manipulation, just what you have to do is to break up the complex exponential term into the sin and cosine and impose the restriction that a i is equal to b i star whereby you will be finding that the imaginary parts they cancel out and what you will be obtaining is real part of a i times cosine pi (2 i plus 1) into (2 n plus 1) divided by 4M. So this is the h i of n so we are getting the so we are now trying to have the h i of n which is nothing but the impulse response of the band pass filter H i what we are realizing; this is the real part of it. So this is out of the multiplication with the real part of a i we get this so this minus i....... the imaginary part of a i that has to be multiplied by the sin of pi (2 i plus 1) (2 n plus 1) divided by 4M; there not only the square bracket ends but also the curly bracket ends and this is multiplied by h of n; remember what is h of n; h of n is the prototype low pass filter response.

Therefore, now just try to interpret this equation, equation number 5 you interpret. Before you interpret this let us have one more relationship mentioned. very simply if you are going to use h i of n to be the impulse response of the filter the ith band filter in that case what will be the response of that filter it will be given by the convolution of the input samples.

Now if you are denoting the input samples as x of n; so the input samples are given by x of n where n is equal to 0, 1,....... etc and supposing we have the length of the filter tap to be equal to......... so L is the length of the filter tap length of the filter tap and we are realizing an FIR filter then the filter output for the bank i so the filter output of bank i this will be given by s l(i) that will be equal to the summation n equal to 0 to L minus 1 x(l minus n) into h i of n simply by the convolution we will be obtaining this so this is the this is equation number 6 that is to say the filter output obtained as a convolution of the ith band pass filters impulse response and the input samples.

(Refer Slide Time: 38:43)



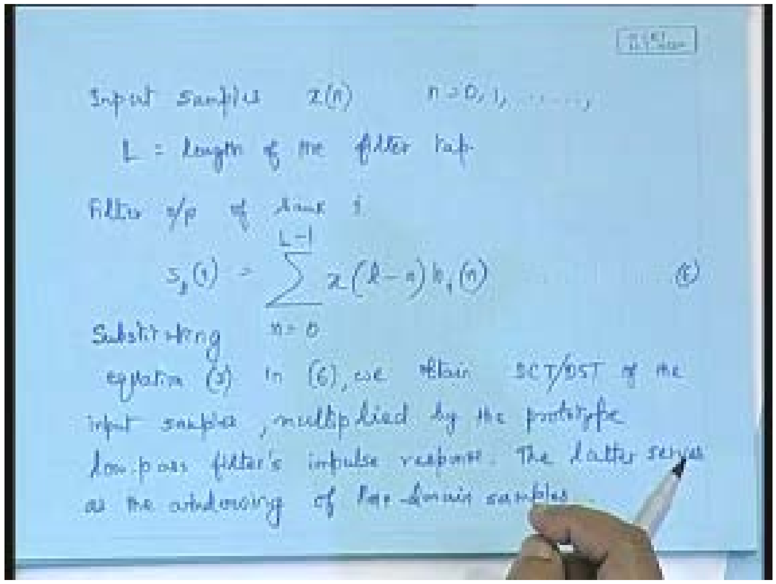
Now, if in equation number 6 you substitute equation number 5 then what you get? Instead of h i of n you write this expression. This expression has got two parts: One is this cosine and the sin part and this is getting multiplied by h of n and if you substitute over here then what you are getting this x or rather to say you are getting xl minus n that gets multiplied by the cosine terms, the sin terms and then you are also getting an overall multiplication by h of n; that overall multiplication by h of n that remains.

Now what is that overall multiplication by h of n?

If you realize the overall multiplication by h of n using a windowing function; windowing is also a kind of convolution. So now if you are implementing that using windowing then what remains to be tackled is x of l minus n or rather the input samples is multiplied by the cosine term or that multiplied by the sin terms. Now these cosine terms whenever you are multiplying the samples by the cosine terms what you are essentially doing is that obtaining the DCT. When you are multiplying by the sin terms what you are doing is the DST; you are doing the discrete sin transform.

Basically whenever you are substituting equation 5 in equation 6, you are obtaining the DCT or DST of the input samples that multiplied by the prototype low pass filters response the prototype low pass filters impulse response and this prototype low pass filter that servers basically as the widowing of the time domain samples. So we can just write down this statement that substituting equation 5 in 6, we obtain the DCT or the DST of the input samples multiplied by the prototype low pass filters impulse response that is h of n. And we can also make a statement that the latter serves as the windowing of time domain samples.

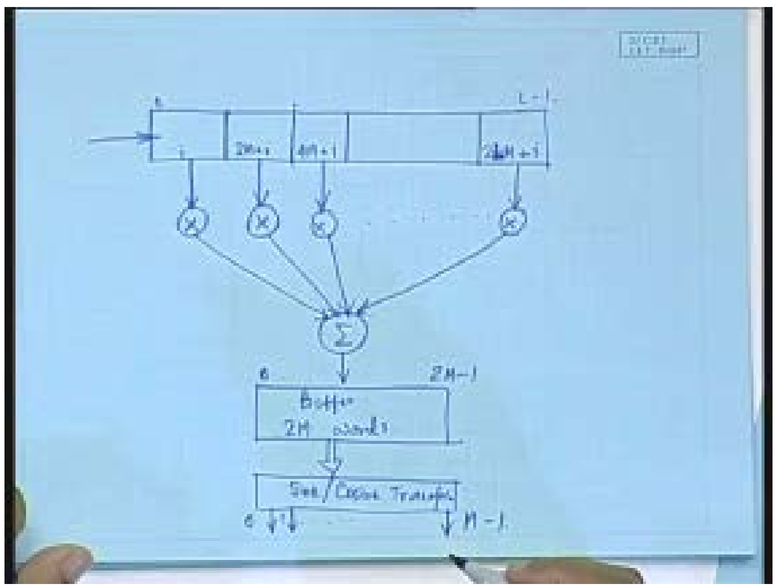
(Refer Slide Time: 42:17)



Now what we can do is that after substituting equation 5 in 6 instead of writing the big equation let us show the block diagram of the implementation. What we do is that one of the efficient implementation that we can talk of; you see that here we are talking of the bank i's implementation. Now there are M such banks, so how to obtain this M banks, so what we have to do is that we can consider that the total length of the buffer we consider an input buffer where we have the input buffer of length L and we have the samples coming in and then we have the sample index as i so we have here 2M samples gathered, the next 2M samples or gathered in this part of the buffer, the next 2M samples are gathered in this buffer so that we index the samples in the second bank as 2M plus i, we index the samples in the third part of the buffer as 4M plus i because up to here we already have 4M samples so the last one will be 2 small m capital M into 2 small m capital M plus i so this is going to be the index of the samples in the last buffer and then what we are doing is that we have to multiply this by the h of n. So, to do the multiplication by h of n............ any doubts or difficulties [Conversation between Student and Professor – Not audible ((00:44:26 min))] 2 LM yes yes not M; yes 2 LM plus i please do the correction I made a mistake and then we have the multiplication by the windowing function so this basically indicates the multiplication by the windowing function and then we are summing them up so there is a summer block and then the summed output is stored in a buffer of 2M words buffer of 2M words and then we apply the sin and the cosine transforms on this.

In fact, applying the sin and the cosine transform together is nothing but applying the MDCT. So this is the sin cosine transform block and there with 2M words as input with 2M samples as input designated as 0 to 2M minus 1 we will be obtaining M sample output from the sin cosine transform block or the MDCT block, so this is 0, 1,........ up to M minus 1.

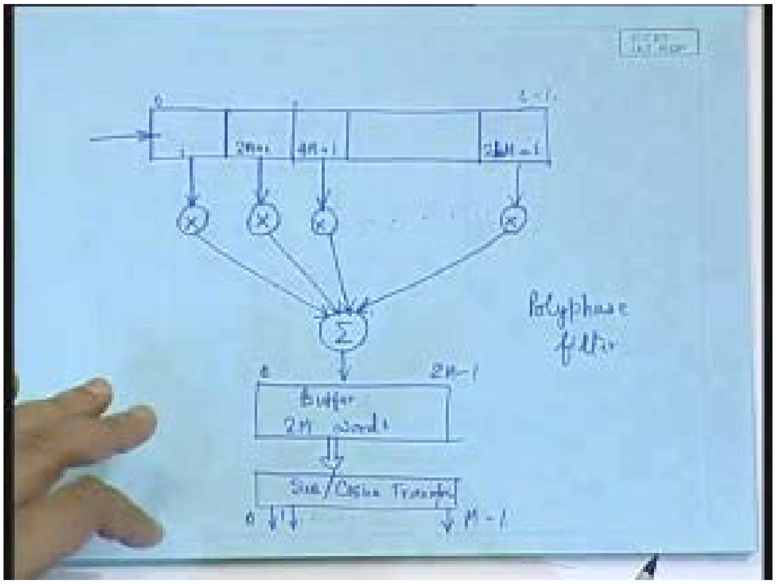
(Refer Slide Time: 46:06)



So these samples are the frequency domain samples what we are obtaining by windowing and sin cosine transform; so the same thing. What we had written down in form of the equation this is just an implementation of this and because this implementation basically involves modulating the center frequency of the filter by a factor of by an amount of pi by 2M or the odd multiples of pi by 2M that is how it realizes a polyphase filter.

You are taking the low pass filter as the basis, the prototype low pass filter as the basis and then by multiplying it with the complex exponentials of the center frequencies; you are obtaining the different filter banks through a polyphase filter technique. And in fact the MPEG audios filtering what I had mentioned in the first place, remember that we started with this diagram (Refer Slide Time: 47:27) so this filter bank in MPEG-1 is realized in a polyphase manner which is very similar to this.

(Refer Slide Time: 47:39)



This is a generalized way of representing but specifically the MPEG-1 audios polyphase filter implementation that goes like this. Please follow this.

In the MPEG audio what we have is that we take an input of 32 samples 32 samples these are in the time domain. Now this 32 samples they will go into where they will be going into buffers and the buffer as you are seeing over here that the buffer is partitioned being partitioned in to all the sub spaces corresponding to the different filters. So we now have this 32 samples that goes into what we refer to as the X FIFO so here we have the first 32 samples, here we have the earlier 32 samples (Refer Slide Time: 48:57) here we have the earlier block of 32 samples like this we have got sixteen previous blocks of 32 samples. So this buffer is of size 512 samples; this buffer stores up to 512 samples so that we have got 16 blocks of such 32 samples.

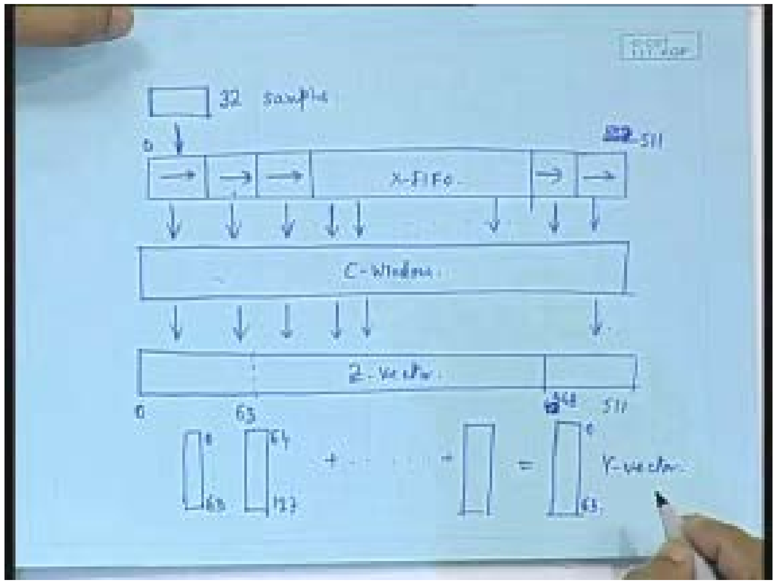
The idea is that, whenever 32 samples arrive this is in the form of a shift register arrangement that is to say that the current 32 samples will be stored over here, the earlier block of 32 samples will move to the next cell, the 32 samples which were there in this cell move to the next and so on so totally we have we are storing 512 such samples and on this 512 samples....... (Refer Slide Time: 49:51) this is 0 to 511 because we have 512 samples so now this X FIFO outputs this will be multiplied by the windowing function and this is called as the analysis window. This will be multiplied by the analysis window and the analysis window is referred to as the C window. So C window is nothing but the analysis window and the windowed outputs they are buffered as I mentioned that they are added and then buffered, so what we do is that these windowed outputs they are collected in a buffer which we call as the Z vector.

Now what is the basic idea?

The basic idea will be that we are feeding 32 samples at the input and we should also have 32 subband samples at the output. Now to get 32 samples (Refer Slide Time: 51:03) that means to say that if I want to make my capital M to be equal to 32 then at the input of the MDCT how many samples I have to feed; I have to feed 64 samples 2M number of such samples so I have to feed 64 samples and we have to then buffer in a buffer space of 64 samples.

What we do is that this Z vector we partitioned into eight such blocks of 64 so 0 to 63 and so on up to 511 so it is partitioned into eight such blocks. So what we do is that this Z vector outputs consisting of 0 to 63, the next block will be 64 to 127 and so on and the last one the last block will be coming from here, this block of 62 so you can calculate this, this will be 424 28 is it 428 or 448 448 so 448 to 511 will be here and they all will be summed up together and there we have the summed vector as the Y vector. So Y vector will be consisting of 64 elements 0 to 63. So all these filter banks so all these individual Z vector components are added. So basically it is a realization of this and this together.

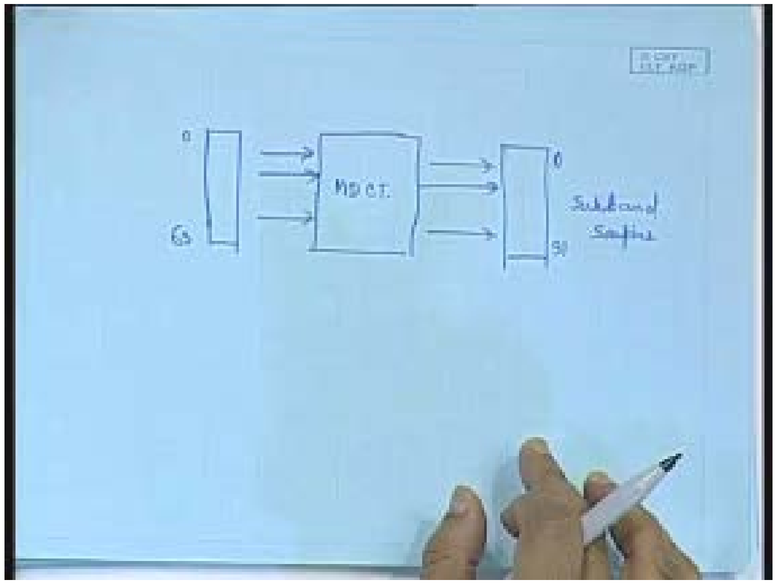
(Refer Slide Time: 52:36)



Therefore, now this buffer of 2M words what we have is nothing but this Y vector. So this Y vector is fed to the......... so this Y vector of 64 samples that is fed to the DCT DST block or rather the MDCT block and this realizes the 32 subbands samples 0 to 31 so these are the subband samples.

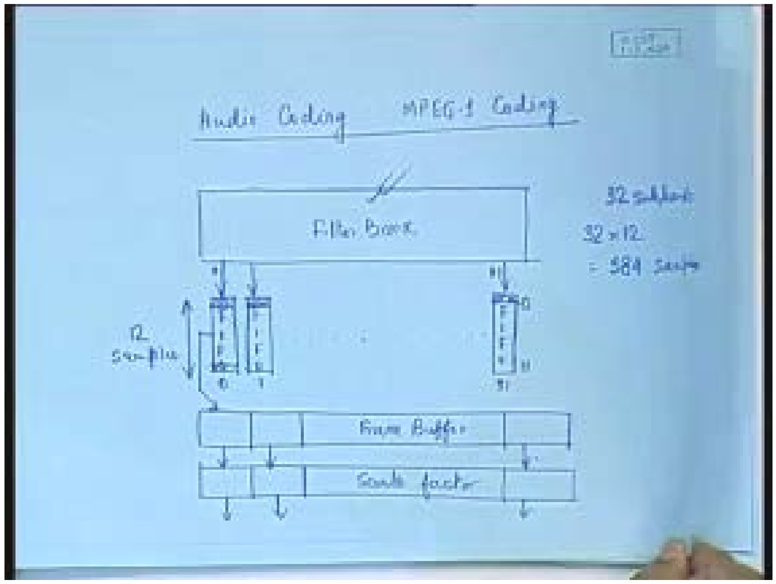
So, for every 32 input time domain samples that you are feeding over here (Refer Slide Time: 53:34) you are getting 32 subband samples as the output from this bank so that sampling in this case is a............. so this is a critically sampled system. For every 32 inputs we are getting 32 outputs and this is the polyphase filter implementation. So what you can see essentially in this implementation again is that we are first passing it through an analysis filter and then we are first passing it through a windowing the windowing is followed by addition and the addition is followed by performing the MDCT and then in the transform domain we are obtaining 32 subband samples.

(Refer Slide Time: 54:38)



This 32 subband samples is what you are going to feed into this first in first out. So 0 to 31 this filter bank outputs will be stored in FIFO, so all the first so all the first set of 31 samples will be stored in the first cell of this FIFO. So when the very first 32 samples arrive they will fill up this block of the FIFO or rather to say the first in first out so the last parts will be filled up and then when the next block of 32 samples arrive the next part of the FIFO will be filled up so that when twelve such set of samples are obtained for each one of the subbands then on this we can do the further processing because in that case we will be obtaining 32 into 12 that is 384 samples which forms an audio frame and on 384 samples we then apply I mean, put it to the frame buffer, perform the scaling, perform the quantization etc.

(Refer Slide Time: 55:46)



It is not only the encoded samples but also the bit allocation information the quantization information, the scale factor information all these have to be put into the channel. So those who are interested to get a more detailed insight about the MPEG-1 audio standard may please refer to this paper this is by S. Shlien same old Shlien and title of the paper is Guide to MPEG-1 Audio Standard. The paper appeared in IEEE transactions on broadcasting; volume 40, number 4, December 1994. The page numbers are 206 up to 218.

So, if you refer to this paper you will get sufficient details. So this completes our discussions on the audio coding and audio transmission. And from the next lecture we will talk about the voice over internet protocol, the VoIP. So, that will be next chapter that we take up from the next lecture, thank you.