Digital Voice and Picture Communication

Prof. S. Sengupta

Department of Electronics and Communication Engineering

Indian Institute of Technology, Kharagpur

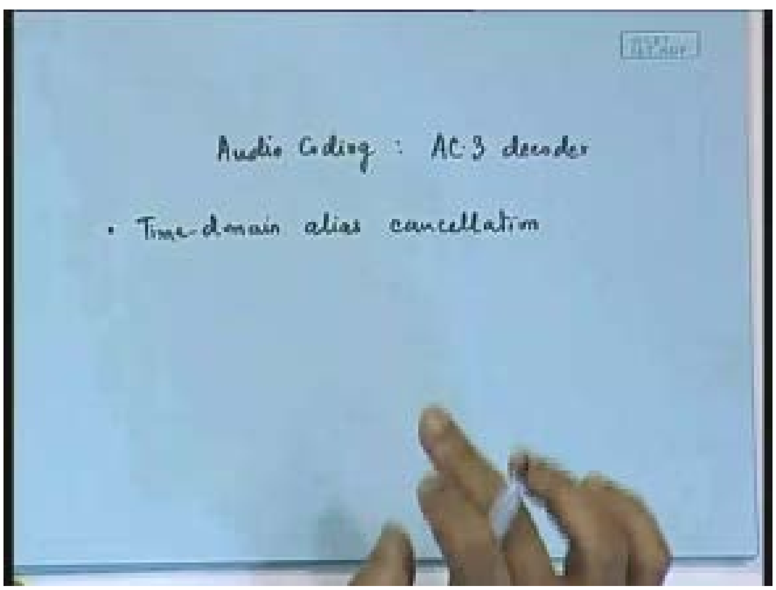
Lecture - 30

AC - 3 Decoder

In continuation with our discussions on audio coding in this lecture we are going to talk about the AC - 3 decoder. In the last lecture we had seen about the basic aspects of AC - 3 and in this one we will specifically look at the details of the decoder that exactly what the decoder has to do and the concept of the MDCT or to say the time domain alias cancellation which I made a bit mention in the last lecture that I will be explaining.

So, before going into the actual topic about the AC - 3 decoder, I will digress slightly and cover the portions that are related to the time domain alias cancellation and realization of the subband filtering through the time domain alias cancellation technique.

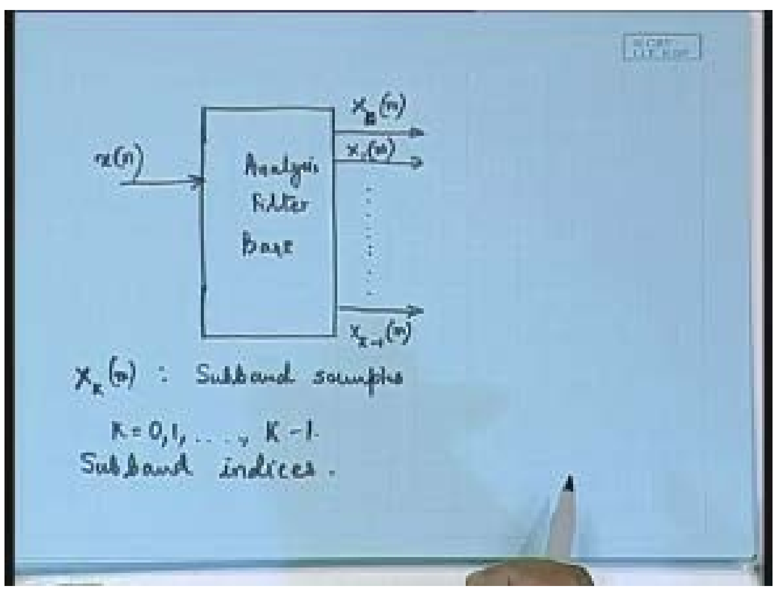
(Refer Slide Time: 2:08)



Now, before we going into that concept let us begin from the basic fundamental aspects of it there let us first take a signal or rather to say discrete samples of a signal which we call as x of n and let us say that we put it through a bank of analysis filters. So we put it through an analysis filter bank so what this analysis filter bank is going to do is to divide these signal into several subbands and individually the subband outputs will be available. So let us denote these by X suffix k of m where (m) is the index of the sequence and this subscript k what we are using is actually the subband number.

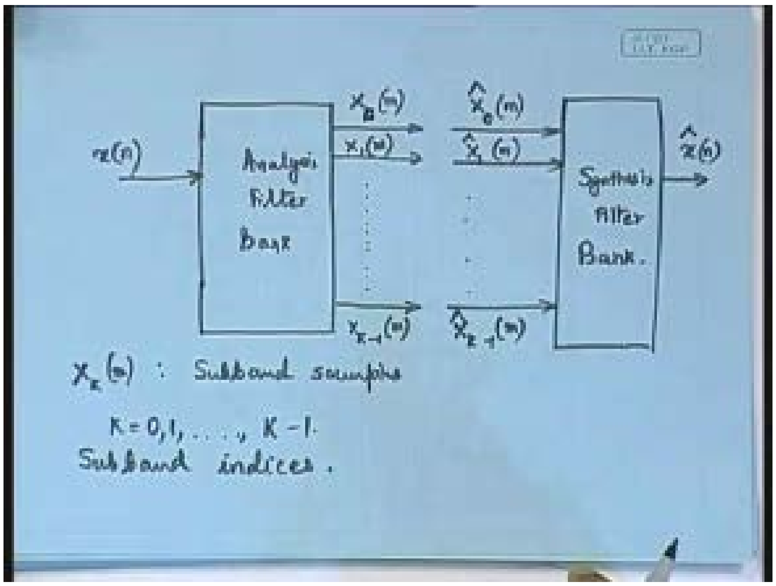
So we will begin with X 0 of m so that the next subband output will be X 1 of m and so on and let us say that we have got k minus 1 up to k minus 1 that means to say that altogether we are having k such subbands so k is equal to 0, 1,............ up to capital K minus 1 so these are the subband indices and this X k of m indicates the subband samples or what we obtain after filtering this x of n through the different subbands.

(Refer Slide Time: 4:17)



So X k of m basically indicates that what is the kth filter response when the signal x of n is put through this analysis filter bank. So, analysis filter bank basically analyses the signal into several such subbands and there are up to capital K number of such subbands which are available at the output of this analysis filter bank. And correspondingly, when we want to recover the signal x of n what we have to do is to put a synthesis filter bank and the synthesis filter bank inputs we are designating as the first one we are designating as X 0 cap (m), the second one we are designating as X 1 cap (m) and the last one we are indicating as X k minus 1 cap (m) and this is put through a synthesis filter bank and the output of synthesis filter bank is going to give us x cap of n.

(Refer Slide Time: 5:41)



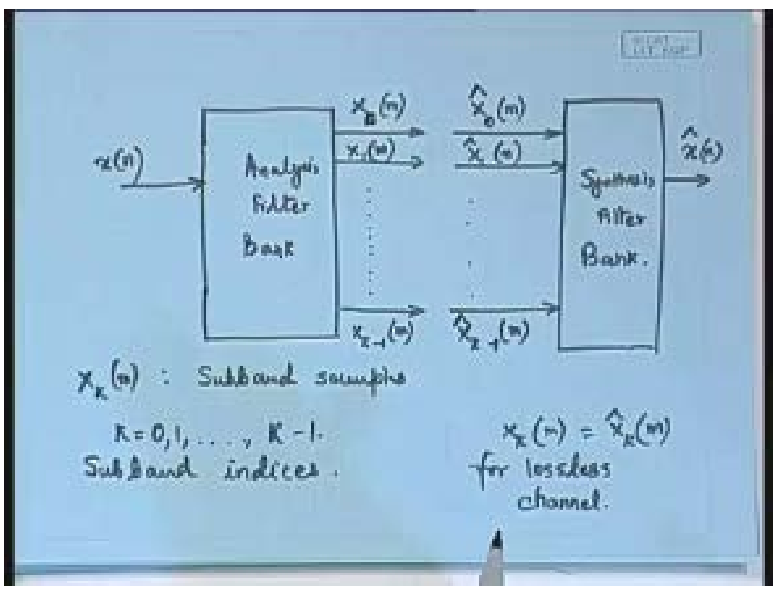
Now, why we are using these caps?

Basically this X 0 2 X k minus 1 these are all the samples, these samples we can encode and they can go into the digital channel and if the channel is noiseless, if it is a lossless channel then X 0 (m) and X 0 cap (m) they become the same. In case of lossless channels we always have X k of m equals to X k cap of m for lossless channels. But even if the channel is lossless there is no guarantee that x(n) and x cap n will be the same because that will basically depend upon how we analyze and how we synthesize.

If the synthesis is done in exact replica to what is being done in the analysis then only we can say that x cap n becomes equal to x of n. but in the process of analysis if any artifact or if any distortion is introduced that distortion has to be correspondingly corrected or that artifact has to be correspondingly corrected in the synthesis only then we will be able to have a perfect reconstruction. So there is definite condition which can be established mathematically for the perfect reconstruction of the signal in the synthesis filter bank.

Now how do we realize this kind of an analysis filter and synthesis filter?
There are two approaches for that. In one approach one can have a bank of filters and in in such kind of cases we will be basically just putting a band of filters and the filter outputs individually the filter outputs will be the same type of samples in the time domain. So we are feeding to the filter bank time domain samples as the input and at the output of the filter bank we will be getting time domain samples of the output.

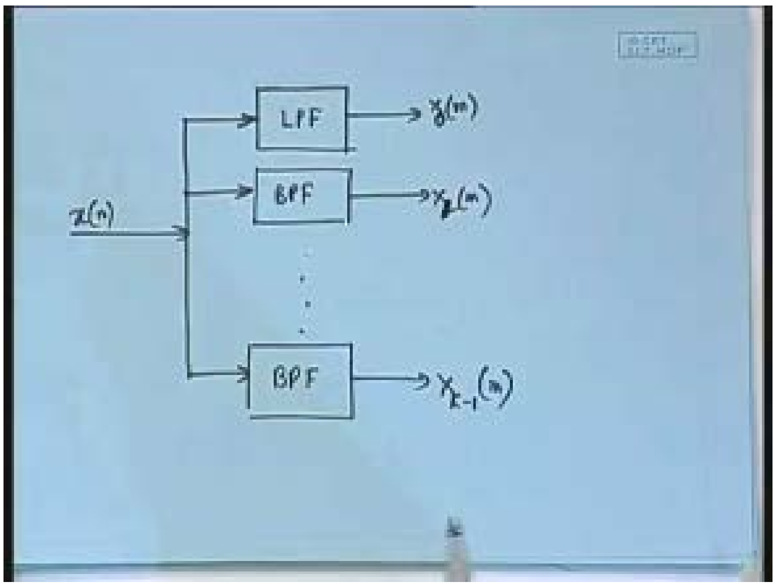
(Refer Slide Time: 7:36)



So basically what we are going to have is that x of n will be fed as an input and it goes to the several banks of filters. So the first one has to be a low pass filter because if this signal x(n) has a bandwidth of let us say 0 to some bandwidth B let us say that the signal is band limited to B and it is 0 to B that is the bandwidth of the signal in that case the first filter that we have to use is the low pass filter; 0 to some low pass cutoff frequency will be some cutoff frequency will be there and then the next bank of filters will be the bandpass filters and the last one will cover a frequency up to B. In fact we have to sample the signal at a rate of 2 if it is band limited to B.

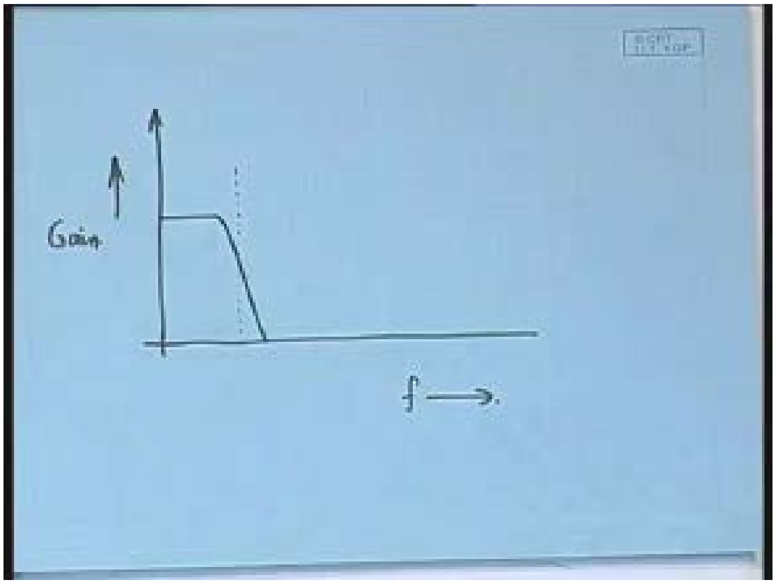
So these are the bank of filters that we realize and individually at this filter outputs we will be observing the signal as I indicated; this will be X 1 (m), this will be X 2 (m) and this will be or X 0 to start with X 0, X 1 and this will be X k minus 1 (m) so this is how we can realize.

(Refer Slide Time: 9:45)



Now individually its frequency responses of the filters will be like this that if we make a plot of the gains of this filter bank, on this axis if we plot the frequency and on the vertical axis if we plot the gain then the first filter bank as I have already mentioned is going to be a low pass filter (Refer Slide Time: 10:22). Now supposing this is the cutoff frequency that we want for the low pass filter, now ideally I should design a filter like this where it goes up to here and then it comes down and then that is zero throughout the rest of the frequencies but I can never make an ideal filter. So my practical filter realization could be something like this so this would be the gain versus frequency plot for the low pass filter.

(Refer Slide Time: 11:12)



Now after the low pass filter we will be having the first bandpass filter. So the first bandpass filter will have its lower cutoff frequency over here. Supposing this is the upper cutoff frequency of the low pass filter and this is the lower cutoff frequency of the bandpass filter so the next bandpass filter will have it is gain versus frequency response like this so this will be for the second filter (Refer Slide Time: 11:44).

Now this is going to have an upper cutoff frequencies somewhere over here so if this is the upper cutoff frequency again the next bandpass will have its lower cutoff frequency over here and it goes up to here, the next band will be having the upper cutoff frequency like this and the lower cutoff frequency for the next filter would be like this. So this is how the responses will be there and the last filter has to be a high pass filter. So, if this is where the last bandpass filter ends then we are going to have a high pass filter beyond this. So if the last stage is a high pass filter the first stage is a low pass filter and in between we have all the filters as the bandpass filters, then this is the kind of gain versus frequency response that is what we will be getting for the individual filters.

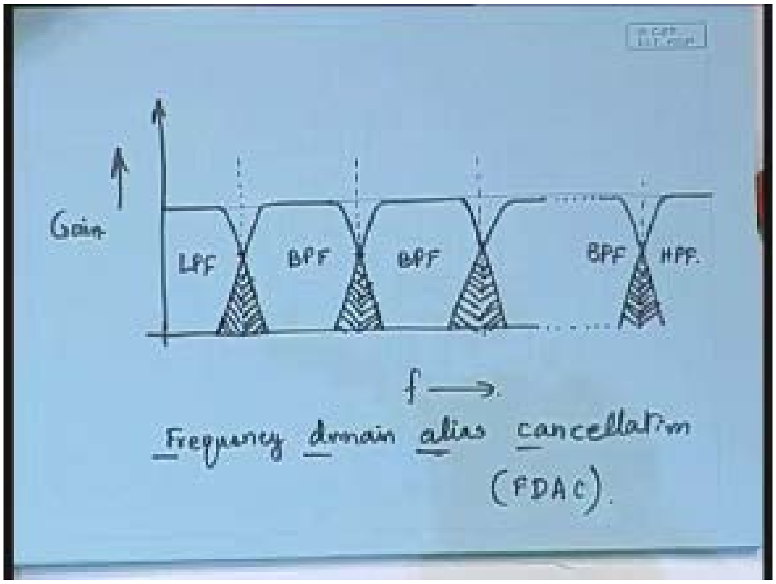
Now the question is that if this is the gain versus frequency plots of the individual filters then is it possible for us to have a perfect reconstruction. If we have an analysis filter bank realized this way (Refer Slide Time: 13:14) and then we put a corresponding synthesis filter bank can we have a perfect reconstruction?

Well, here (Refer Slide Time: 13:27) if you see the frequency gain versus frequency responses you can see that after all what we are going to achieve for a perfect reconstruction there should not be any frequency distortion of the signal. That means to say that if we add up the responses of these individual filters then we should be getting a flat frequency response throughout this thing, we should be getting a flat frequency response, so, in order to get a flat frequency response what is the condition, that, one has to fulfill.

You see, look at the low pass filter. We are saying that this is the cutoff frequency. if this is the cutoff frequency in that case what are these components what are these frequency components. These frequency components are then the alias components. So the alias components will be appearing this way. Similarly, for this filter the alias components of here will be appearing this way so that the response that we get in this zone they are the aliased filter responses these are the aliasing component. Likewise here also we will be having the aliased components.

Now, in order to get a flat response what should we do?
We should have a proper alias cancellation during the synthesis. If we add up and then we have an alias cancellation in the synthesis process then only we can get a flat frequency response. So the alias cancellation that we are doing in this case is a frequency domain alias cancellation; in short form we can call that as FDAC which stands for frequency domain alias cancellation.

(Refer Slide Time: 16:07)

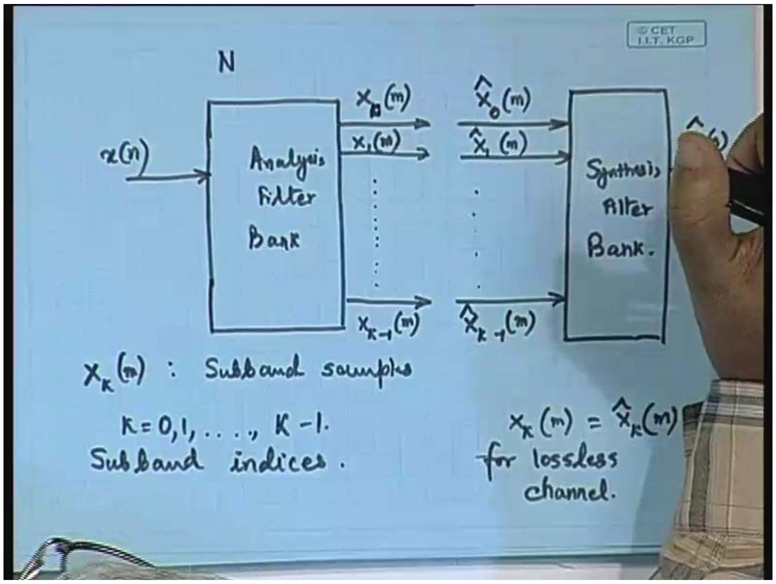


Now look at these characteristics. What you find is that in terms of the frequency responses of the individual filters there is an overlap that is present. can you see the overlap; because unless the overlap is there.................. see first of all that we could not realize a perfectly rectangular response filters that is ruled out we cannot practically achieve that so that is why a practical filter has to have this kind of a characteristic and that is why in terms of its frequency response also there is an overlapping that exists in its frequency bands.

So now, overlapping in the frequency domain characteristic and realizing in frequency domain alias cancellation that is exactly what we do if we realize the analysis filter bank in this manner (Refer Slide Time: 17:14) having individual low pass and high pass filters. Now here all the processing is being done in the time domain but the alias cancellation that we are doing is a frequency domain alias cancellation but rest of the processing everything is in the time domain.

Now let us look at an alternative way of realizing this analysis and synthesis filter.

(Refer Slide Time: 00:17:47)



In this case what we want to do is to have the filtering in the transform domain; rather than in the time domain we would like to have a transform domain filtering; how, let us say that we feed this signal x of n the samples and these samples are first windowed; we apply a window whether rectangular window or some other window some window we apply some windowing function we apply on to it and then the windowed version of the signal let us put through a transform say DCT DFT some kind of a transform we do and then the transformed signals is what we get. These transforms will basically correspond to the individual subbands follow my point. That means to say that if you are looking for the low frequency part in that case where will you get this.

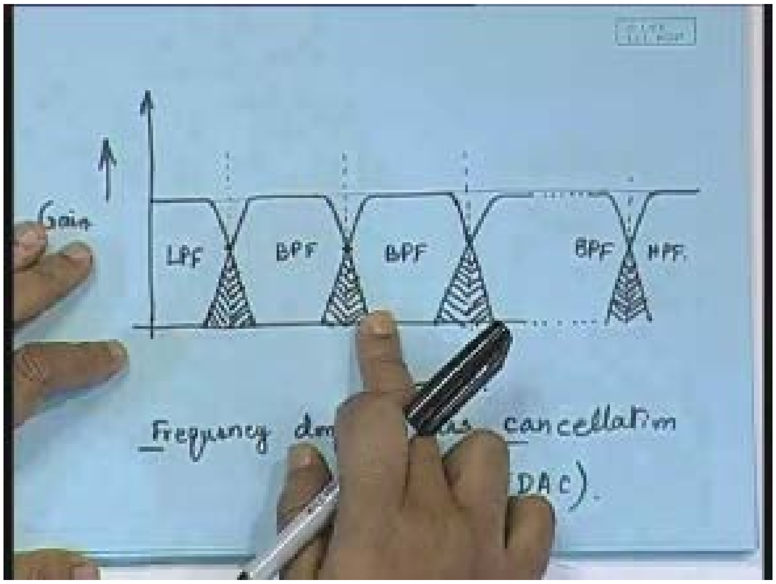
In the transforms supposing you take N number of samples you take capital N number of samples and you obtain an N point DCT you window N number of samples and then on windowed N number of samples you apply a DCT so then the DC part of the DCT that means to say that the very first coefficient if you look at it.................. so you are getting the very first coefficient as the DC or the lowest frequency and here you will be getting the highest frequency component.

Now, when the next frame of x(n) arrives you again perform a windowing and then again you will be observing the lowest frequency here highest frequency here. So if you continuously monitor this what will you get all the lowest frequency samples; if you continuously monitor this what will you get all the highest frequency samples so this is also a filtered realization, only thing is that here we are first having a time domain windowing and the time domain windowing is followed by a frequency domain transformation.

Now, in the frequency domain transformation the signals that you are getting at the analysis filter output they are not in the time domain but they are now in the frequency domain. So what you have to do?

You send these frequency domain samples into the channel assume that the channel is lossless as before then this frequency domain samples now you put through an inverse transformation, you apply the inverse transform of exactly what you had done in the analysis filter and the inverse transform output that you now get.......... at the inverse you are getting it in the time domain but again you have to compensate for this windowing. And in this case what we can do is that because we have to compute the transforms in a continuous manner the window also should be an overlapping window. If we talk of overlapping window in that case we can apply a duality of what we had as an overlapping window like this.

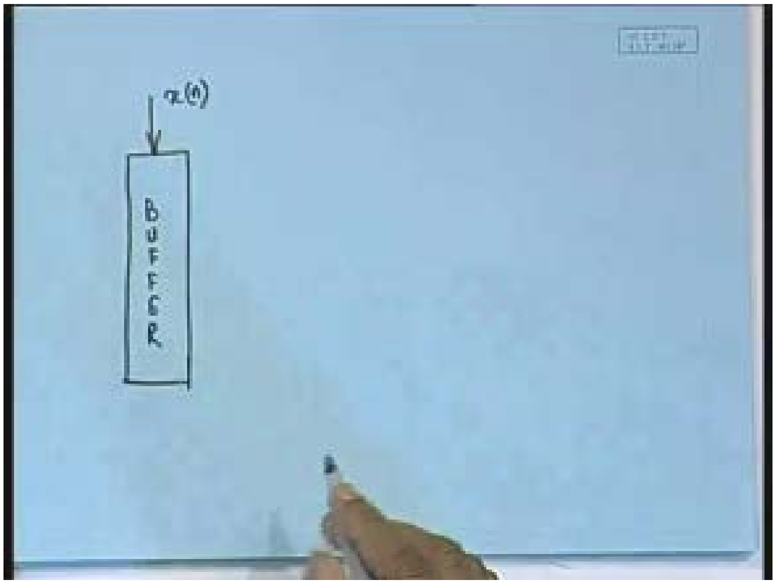
(Refer Slide Time: 22:02)



In this case this also is a window but this is a window in the frequency domain. But likewise if we apply a windowing in time domain and if those time domain windowing is also an overlapped one in that case what we are doing is that on the signal we are applying overlapped time domain windowing taking a transform of those at the synthesis part taking inverse transform of this and then on these windows the corresponding synthesis has to be applied which has to be done in a manner of overlap and add technique.

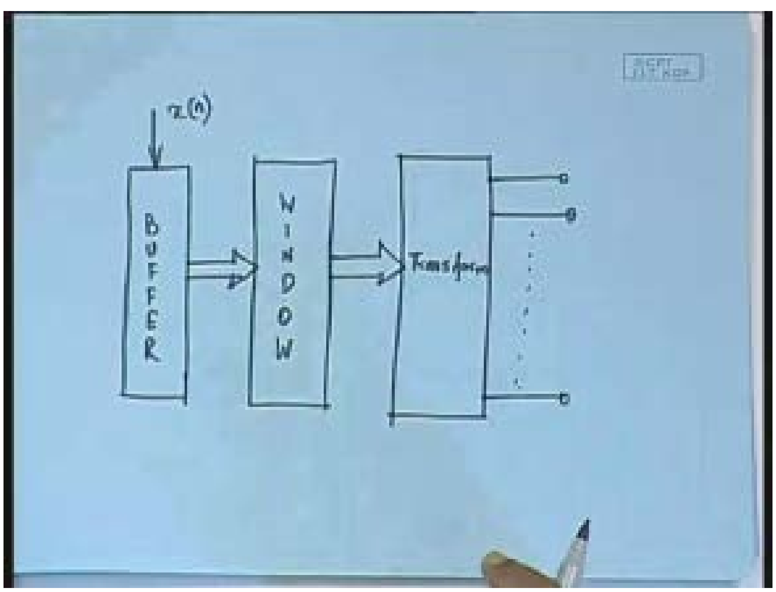
Now in this case while doing the overlap and add technique what did we achieve; we achieved frequency domain alias cancellation. Now in this scheme what we are going to achieve; the windowing is in the time domain, so mind you what we are doing is that this x of n these are the samples which are say arriving in a buffer.

(Refer Slide Time: 23:27)



Supposing we are storing the incoming samples in a buffer and this buffer contents are put through a window; so this is a window that is applied on the buffered samples and then on this windowed samples what we do is we apply transform and the transform outputs are this which we take as the subband signals.

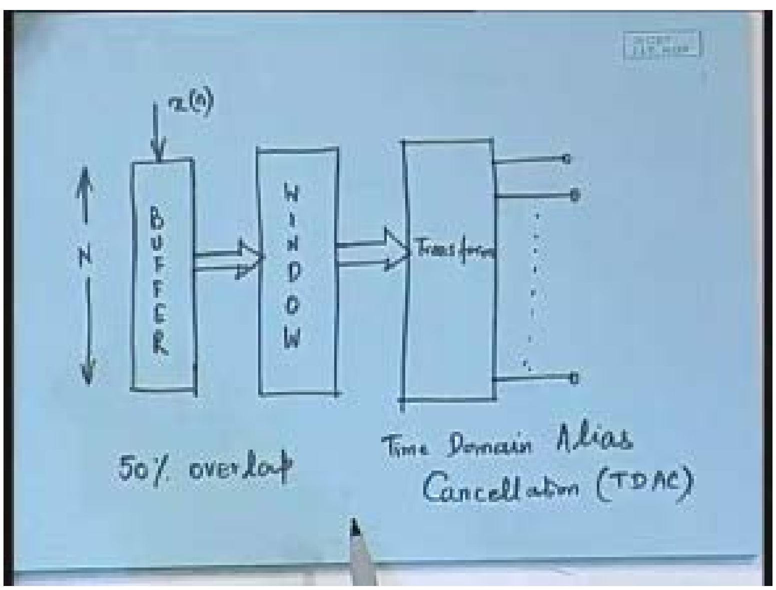
(Refer Slide Time: 23:59)



Now this buffer what we are having is an overlapping kind of a buffer meaning that it is not that here we store N samples, take the window and then take the next N samples and take the window not like this; rather there is a degree of overlap and the degree of overlap is 50 percent overlap is applied. So 50 percent overlap if you want to achieve what you have to do is that once you take a windowing on N number of samples in the first frame then what you have to do is that in the next frame you shift N by 2 samples out of it and let N by 2 number of new samples arrive. So you are having N by 2 number of new samples, you have N by 2 number of old samples and then you apply the window.

In that case what you are doing is you are having the windowing but with an overlap something very similar to this (Refer Slide Time: 25:15). But this you did in the frequency domain and this you are doing in time domain (Refer Slide Time: 25:18). So, in the process of synthesis what you want to achieve is time domain alias cancellation in short form this is called as the TDAC.

(Refer Slide Time: 25:40)

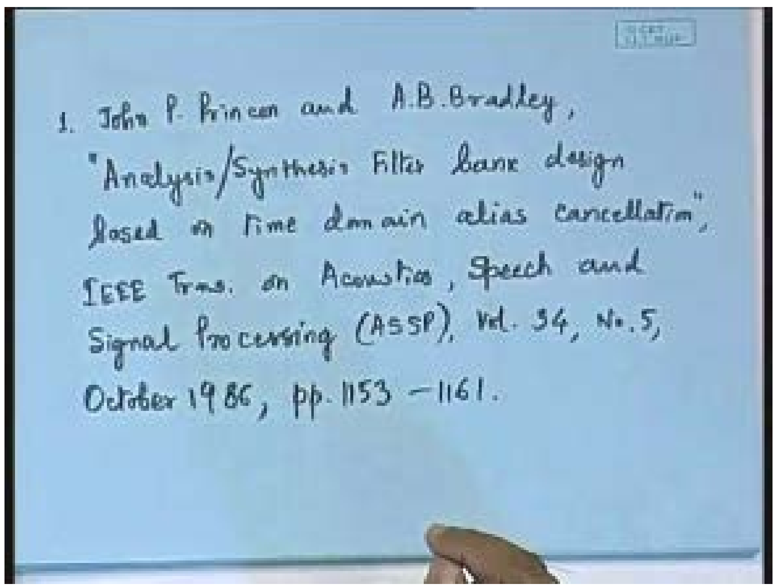


Now this I have only qualitatively described and rather tried to explain this to you conceptually but there is a very involved and nice mathematical treatment which is available and I would advise you to go through this reference. The reference is like this: it is by John Princen John P.

Princen and A. B. Bradley and this is analysis synthesis filter bank design based on time domain alias cancellation.

This paper appeared in IEEE transactions on Acoustics, Speech and Signal Processing, in short form ASSP Volume 34: Number 5, October 1986, page numbers 1153 to 1161. So you can consult Princen and Bradley's paper. They have analytically developed the theory and analytically they have proved that how the time domain alias cancellation could be achieved.

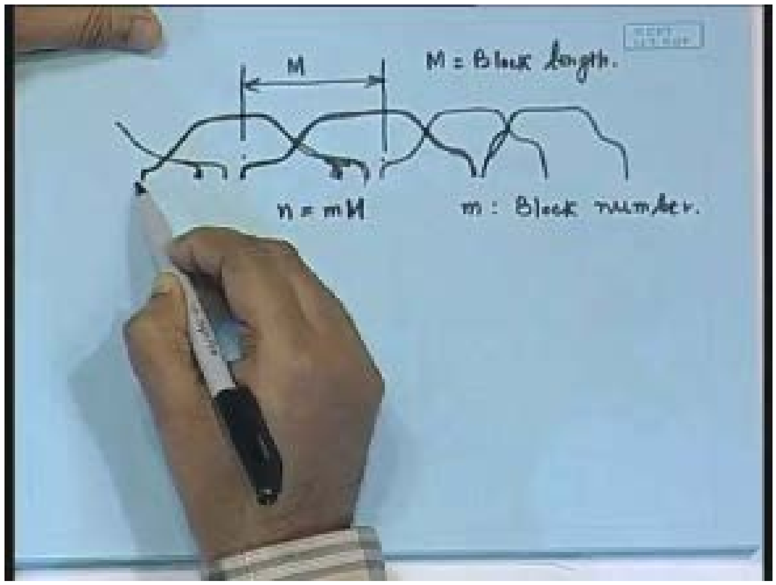
(Refer Slide Time: 27:45)



But I described the same thing to you qualitatively. And now let me tell you in a block diagram form that how we are going to achieve that.

The time domain aliased cancellation based on the analysis synthesis filter this is achieved like this that supposing we have the input signal and let us say that we have the input signal like this, say this is the window that we are applying; not my drawing is not very now what I mean to say is that supposing this duration from here to here means in between these windows we are having (m) number of samples. Now in this case (m) is our Block length; capital (m) is our Block length and we have the sample index as n equal to small (m) into capital (m) where small (m) happens to be the block number.

(Refer Slide Time: 30:20)

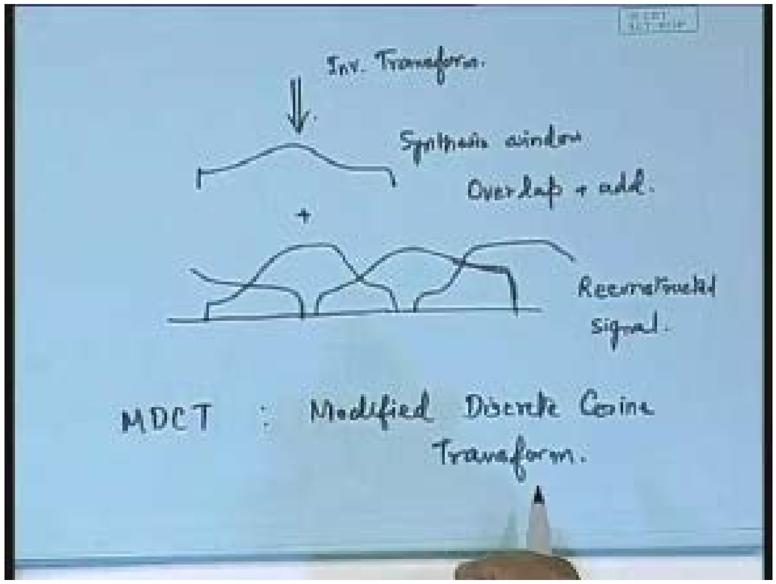


Now what I mean to say is that this is one block of samples, this is the next block of samples so can you see the overlap, this block of samples has got this much. Now this is a windowing function so it is purposefully drawn so as to appear as non-rectangular window. So you can see that this is a kind of windowing that we have applied over here and the next block's window is like this (Refer Slide Time: 30:38) which contains some of the samples of the earlier window also; at least 50 percent of the samples of the earlier window is also there and then some new samples which are also coming in.

Then again the next window goes from here to here; the next window goes from here to here like this so these are the windowed version of the signals and then on this we are applying transforms so these are the analysis filtering part so this is the analysis part and then the analysis output will be X 0 (m) to X k of m; mind you these are in the frequency domain k is equal 0, 1,.......... up to capital K minus 1 and here we have the inverse transformation, this is the inverse transform and this is synthesis part and then the inverse transform output what we do is we overlap and add, so the overlap and add is actually done using a window like this so the inverse transform output so these are the inverse transform output so they are put through a synthesis window the synthesis window is like this and the synthesis window we apply an overlap and add, so this is the synthesis window (Refer Slide Time: 32:50) and the overlap and add is done in order to correct the time domain aliasing which is introduced so that the reconstructed signal would then be like this that whatever we have got as the windowing so here we will be getting the reconstructed signal at the output of those overlap and add blocks, so the synthesis has to take care of this in order to cancel the time domain aliasing.

Now this is the basic concept. And in fact, the transform what is being done is actually a DCT but when you are taking an overlapped DCT on these overlapped samples; that means to say that DCT plus this kind of a 50 percent overlapping window this compositely realizes what is called as the modified discrete cosine transform, MDCT which is the modified discrete cosine transform which is the DCT applied over this overlapped window.

(Refer Slide Time: 34:39)



Now we go back to our AC - 3 codec and see that how this concept is utilized, I mean, how we are making use of this MDCT block in the AC - 3 decoder. Let us recollect the frame structure of the AC – 3.

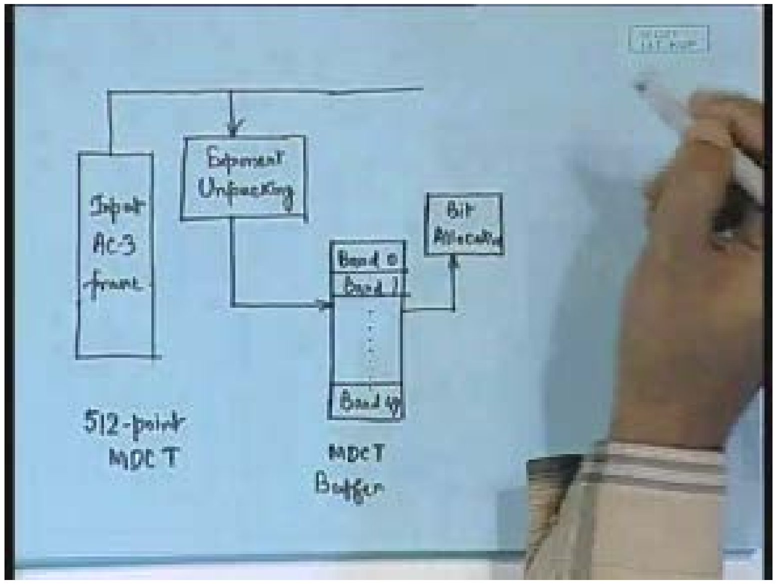
In the frame what are we having? In this part we have the input AC - 3 frame (Refer Slide Time: 35:17); so the input AC - 3 frame will be containing the exponent and the mantissa information corresponding to all the channels that it is encoding. So what you have to do in the first place is to do the exponent and unpacking.

Actually exponent unpacking you can do directly because exponent unpacking is not dependent upon the bit allocation information. You do not have to extract the bit allocation information. In fact what will be done is based on this exponent information the bit allocation information will be generated by the decoder itself. So what we do is that from the AC - 3 frame we extract the exponent part so this is the exponent unpacking and then we will be putting these exponents; all these extracted exponents will be put in to an MDCT buffer so this we are calling as the MDCT buffer and this MDCT buffer is actually........... what is it containing; it is containing the frequency bands because already these samples what we are receiving here they are all frequency domain samples, we have to do an inverse transformation before we actually obtain the reconstructed signal but so far we are dealing everything in the MDCT domain so all the exponent unpacking what we obtain here are all the MDCT coefficients so the MDCT coefficients will be put in to a buffer and in fact the MDCT which is applied in the AC - 3 encoder is a 512.0 MDCT.

In fact in AC - 3 this 512.0 MDCT can be switched into two 256 points in MDCTs by what is called as the switched bank technique and the switched bank technique is applied whenever any transient is detected in the audio frame; when any transient is detected then the 512.0 MDCT will be split into two 256.0MDCT by the switched bank MDCT computation but all that I mean to say here is that after the exponent unpacking it goes into the MDCT buffer. That means to say that individually all these components will indicate the corresponding frequency bands.

Now totally we have, I mean, going by the critical band philosophy in order to extract the masking information the auditory spectrum, I mean, according to the AC - 3 decoding it is divided into fifty such critical bands. So we have the MDCT buffer information segregated into this fifty bands so band 0, band 1 etc up to band 49 so all the so the MDCT coefficients this 512.0 MDCT coefficients will be divided into fifty such bands and using these bands, I mean, using the coefficients which we obtain from these banks, the bit allocation information can be derived by using the psychoacoustic model. So the bit allocation will be decided over here.

(Refer Slide Time: 40:13)

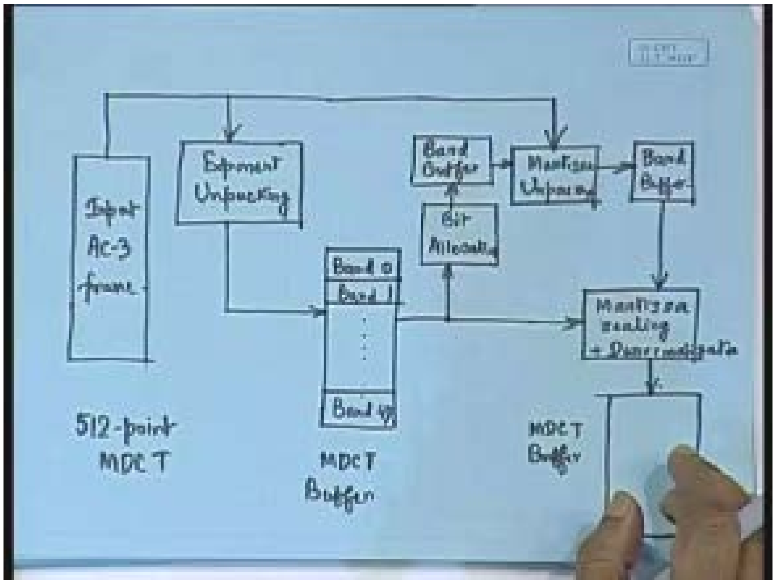


Now, at the same time the input AC - 3 frame we are not only deriving the exponent but also we are deriving the mantissa. So we do the mantissa unpacking so this is one block where we do the mantissa unpacking and then the bit allocation information that goes into a band buffer because the bit allocation will be pertaining to a band. So there is a band buffer that contains the bit allocation information for that particular band and that controls the mantissa unpacking because the mantissa remember was encoded based on the bit allocation information so it has to be decoded also by consulting with the bit allocation. So, for that particular band whatever bit allocation information is extracted that will decide that how many mantissa bits will be extracted from the bit stream and accordingly the extracted mantissa will go into a band buffer.

Therefore, mind you, the purpose of these two band buffers is different. This left hand side band buffer stores the bit allocation information whereas the right hand side band buffer that stores the mantissa information the extracted mantissa information. Now here we are getting the exponent information already stored in the MDCT buffer. Now we require both the exponent as well as the extracted mantissa information.

So what we do is that the output of this band buffer is combined with the exponent information and we perform mantissa scaling and denormalization. So this is where we do mantissa scaling plus denormalization and then this (Refer Slide Time: 42:45) mantissa scaling and denormalization information is actually stored into the MDCT buffer. This MDCT buffer will be actually replaced by this so this is also MDCT buffer so this earlier information which contained only the exponent will now be replaced by the information the complete information of exponent plus mantissa into this MDCT buffer.

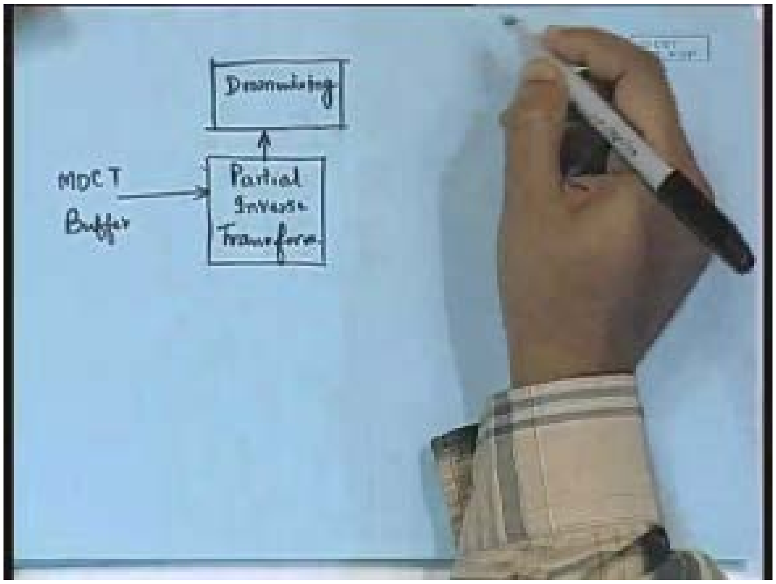
(Refer Slide Time: 43:22)



So now what we are doing; you see, here we could not do any inverse transformation that was not possible because we only had the exponent information. But now in this at the MDCT buffer here we are getting both exponent and mantissa information. So now this information we are able to obtain the inverse transformation of this. So we apply the inverse transformation but the complete inverse transformation process should involve the inverse transformation plus the overlap and add. Therefore, what we do in this case is the overlap and add part is tackled separately and instead we just take the inverse transformation and because the overlap and add part is avoided we call that as the partial inverse transform. So this MDCT buffer output from here will be taken like this.

The MDCT buffer output the MDCT buffer output that contains the exponent as well as the mantissa information, this goes into a block which does the partial transformation partial inverse transformation partial inverse transform and this goes into a block which performs the down mixing.

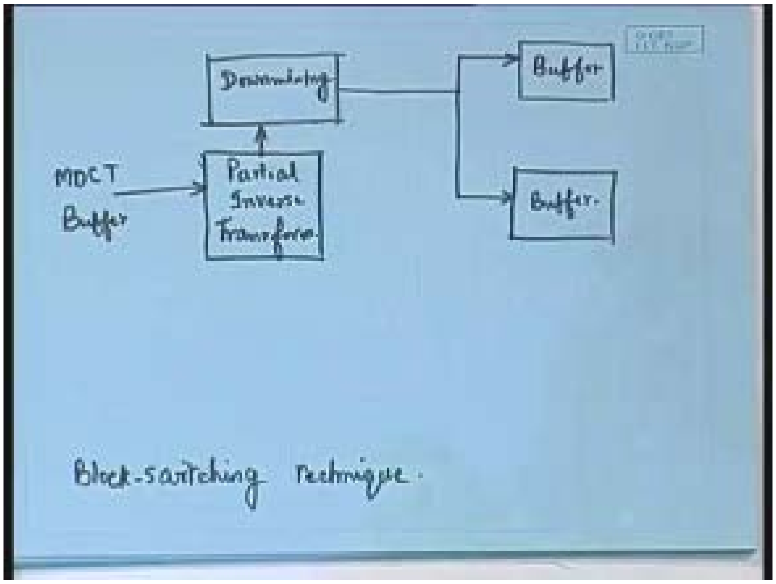
(Refer Slide Time: 45:10)



Now I was telling you about down mixing in the last lecture. basically what we mean to say is that the number of channels that we are using at the encoder and the number of channels that we use at the decoder they need not be the same so that is why this inverse transformation information is put through a down mixing which stores the down mixed information into two buffers.

Actually speaking, this is also to realize the switched bank information so what I was telling you little while back. That means to say that two 256 point transforms are computed so that is the block switching technique. So it follows from the block switching technique. I am not into the details of this technique because the details of this technique are not easily available since it is a trademark of Dolby Corporation so some information is not a fully disclosed. But to achieve this block switching technique what is being done is that this inverse transform output after being down mixed is split into two buffers and we call this as down mix buffer 1 and down mix buffer 2.

(Refer Slide Time: 46:53)

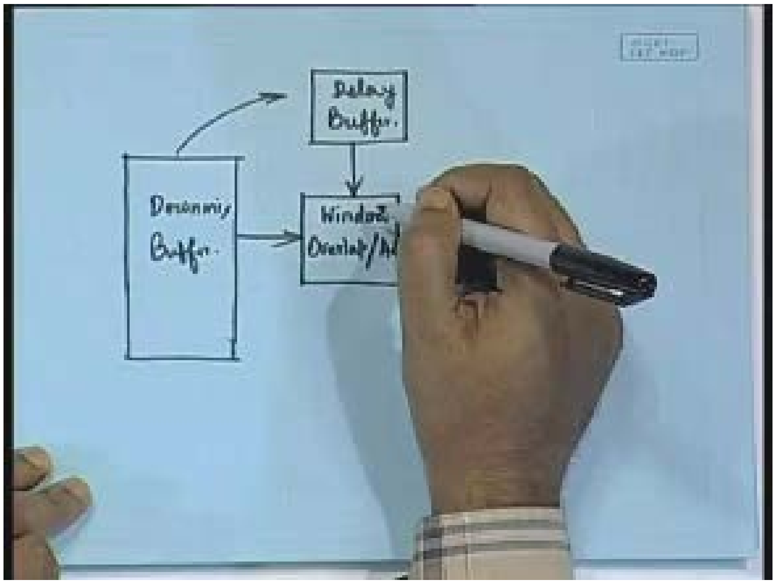


So this whole thing means to say.......... starting with this input AC - 3 frame (Refer Slide Time: 46:59) and storing the partial inverse transform information into this down mix buffer this entire thing realizes what is called as the decoder input processing. Mind you, this has not realized the PCM signals back because ideally because what we want to achieve at the decoder is that we want to get back the PCM samples.

But so far we have not obtained it because we have only stored it into the buffer but we have not yet obtained the PCM output. So what we have to do is that this decoder input processing what consists of all these blocks: exponent, unpacking, storing into the MDCT buffer, mantissa unpacking, doing mantissa scaling and denormalization, obtaining the exponent and mantissa part and then doing the partial inverse transformation and down mixing this whole process is the decoder input processing and the decoder input processing is followed by a decoder output processing.

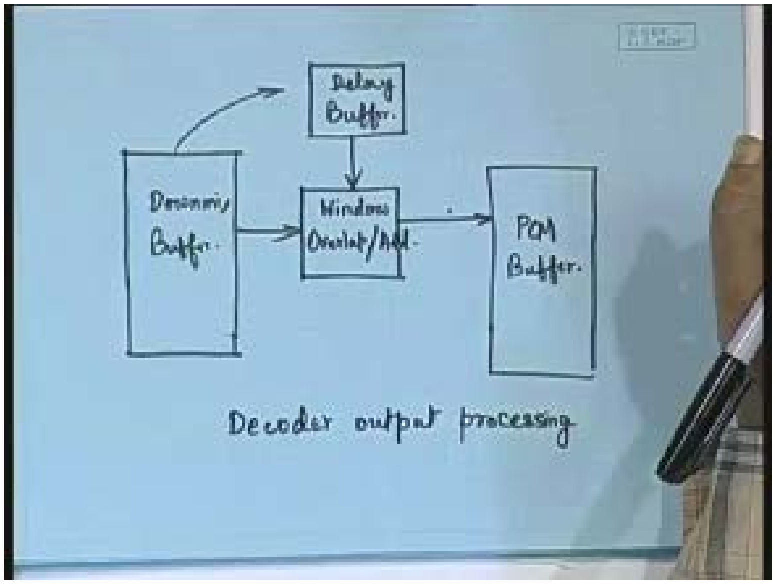
In decoder output processing what we do is that we start with down mix buffer and then the down mix buffer contents are..................... actually we not only take this down mix buffer but also by using a shift register like arrangement we have a delay buffer and delay by how much; delay by half of the frame times. The purpose? Because we are having a down mix buffer and also a delayed version of the down mix buffer that is why we can have a 50 percent overlap.

(Refer Slide Time: 49:35)



So in this block what we are doing is that the delay buffer and the down mix buffer contents they are window add and window overlap and add. So we apply window overlap and add on this down mix buffer and delay buffer content and the output of this window that goes into the PCM buffer. This is where (Refer Slide Time: 49:47) we ultimately achieve the inverse transformation because the window overlap and add is computed over here so this is where the PCM samples are obtained back. So these blocks combinedly we call as the decoder output processing.

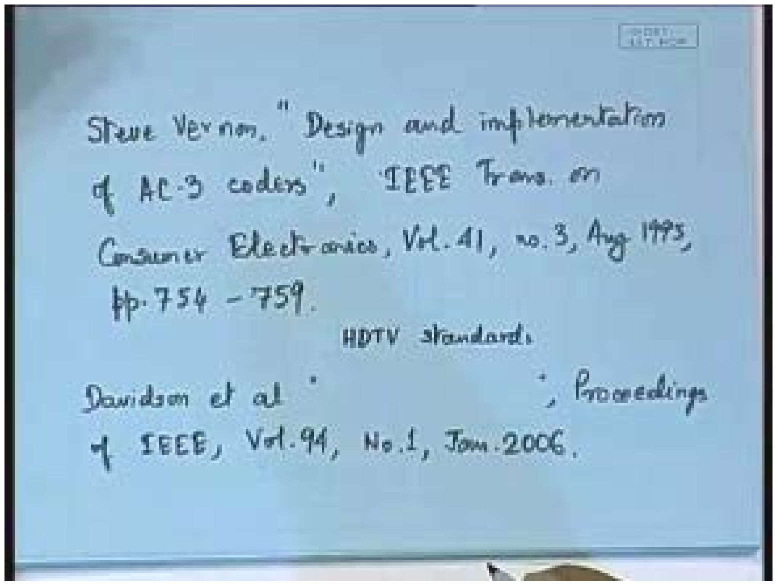
(Refer Slide Time: 50:21)



Therefore, you can see that essentially the decoder has got two parts: one is the decoder input processing which we described a little while back and the decoder output processing which I described just now. When the down mix buffer is of size 256 then the delay will be done by 128 samples and then we are having a 256.0 MDCT computed or rather 256.0 inverse MDCT. Hence, if the block switching flag; okay, remember that in the bit-stream we also mentioned about certain flags and one of the flag is a block switching flag. So if the block switching flag is there in that case we are going to have the down mix buffer of size 256 and then we are obtaining a 256.0 inverse MDCT and obtain the samples this way.

Whereas if no transient is present in that case we will be having 512.0 inverse MDCT and how to obtain that; we will be picking up the two down mix buffers together and apply a delay of 256 and have this. So this arrangement of having the down mix buffer really permits but how exactly the switching is done that secret is not known to me. And to know little bit more about what I have already described; in fact the material that I covered most of it has been taken from one very interesting paper that is written by Steve Vernon, the title of the paper is Design and Implementation of AC - 3 Coders and this appeared in IEEE transactions on consumer electronics; Volume 41, number 3, August 1995. The page numbers are 754 to this is not a very long paper 754 to 759 and subsequent to this......... no not much of very extra information has been added to it.

(Refer Slide Time: 54:51)



There is a paper by Davidson, I do not have the complete details with me, may be in the let me let me see, yes, there is another paper which you can refer to, that is by Davidson. Davidson I write as et al' because i do not have the information about the co-authors exactly and this paper's title also I cannot............ I will tell you tomorrow. It appeared in proceedings of IEEE proceedings of IEEE that is Volume 94 number 1, it is a very recent paper appeared in January 2006 and this is a paper on the............. if I remember the title correctly it is on the HDTV standards or something. This is a paper on the HDTV standards.

Do not remember the title of the paper exactly. And in this Davidson et al' not only covered the video standard for HDTV which is as I mentioned is the MPEG-2 but also they covered the AC - 3 which is the audio coding standard that has been followed in the HDTV. So this gives almost similar kind of details may be with a few more examples etc which you can go through. So I would advise that you go through both these papers to get a better feel of the AC - 3 coders.

In the next class I will be making a very brief mention about the MPEG audio coding standards and then we will go over to the new topic, thank you.