

# ON Semiconductor

## Is Now

# onsemi™

To learn more about onsemi™, please visit our website at  
[www.onsemi.com](http://www.onsemi.com)

---

**onsemi** and **onsemi** and other names, marks, and brands are registered and/or common law trademarks of Semiconductor Components Industries, LLC dba "**onsemi**" or its affiliates and/or subsidiaries in the United States and/or other countries. **onsemi** owns the rights to a number of patents, trademarks, copyrights, trade secrets, and other intellectual property. A listing of **onsemi** product/patent coverage may be accessed at [www.onsemi.com/site/pdf/Patent-Marking.pdf](http://www.onsemi.com/site/pdf/Patent-Marking.pdf). **onsemi** reserves the right to make changes at any time to any products or information herein, without notice. The information herein is provided "as-is" and **onsemi** makes no warranty, representation or guarantee regarding the accuracy of the information, product features, availability, functionality, or suitability of its products for any particular purpose, nor does **onsemi** assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation special, consequential or incidental damages. Buyer is responsible for its products and applications using **onsemi** products, including compliance with all laws, regulations and safety requirements or standards, regardless of any support or applications information provided by **onsemi**. "Typical" parameters which may be provided in **onsemi** data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. **onsemi** does not convey any license under any of its intellectual property rights nor the rights of others. **onsemi** products are not designed, intended, or authorized for use as a critical component in life support systems or any FDA Class 3 medical devices or medical devices with a same or similar classification in a foreign jurisdiction or any devices intended for implantation in the human body. Should Buyer purchase or use **onsemi** products for any such unintended or unauthorized application, Buyer shall indemnify and hold **onsemi** and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that **onsemi** was negligent regarding the design or manufacture of the part. **onsemi** is an Equal Opportunity/Affirmative Action Employer. This literature is subject to all applicable copyright laws and is not for resale in any manner. Other names and brands may be claimed as the property of others.

## WOLA Filterbank Coprocessor: Introductory Concepts and Techniques



ON Semiconductor®

<http://onsemi.com>

### APPLICATION NOTE

This tutorial is applicable to: Toccata Plus™, Orela® 4500 Series, BelaSigna® 2xx Series

This Application Note details the concepts of WOLA filterbank processing and the configuration of the WOLA filterbank coprocessor for any application. For the sake of consistency, the example of a hearing aid application is used throughout to demonstrate these concepts. However, the WOLA filterbank coprocessor can be configured creatively within its constraints to benefit a wide variety of audio and other applications.

#### INTRODUCTION

This tutorial is dedicated to the WOLA filterbank coprocessor, which is integrated in the SignaKlara® DSP architecture. The goal of this document is to address the theoretical aspects of the WOLA filterbank, and describe the influence of each involved parameter. A good level of understanding at signal processing level is required in order for the user to properly select the appropriate filterbank configuration in a specific application. Providing basic concepts, extensive examples and illustrations, this text guides the user in this task. The tutorial is organized in the following manner:

##### Signal Processing Aspects:

- Multirate filterbanks: the tree-structured decomposition, an introductory example
- Complex-modulation filterbank: original and complex-bandpass interpretations
- Weighted overlap-add implementation: the WOLA
- Prototype filter design

##### Filterbank Design: Playing with the Filterbank Parameters (Using the MATLAB® WOLA Toolbox)

- Typical hearing-aid configurations (16 bands)
- Configurations with refined frequency resolution
- Overlap-add configurations (short-term discrete fourier transform)
- Performing FFTs with the WOLA filterbank
- Using special microcodes

#### SIGNAL PROCESSING ASPECTS

This section introduces the basic signal processing principles involved in multirate filterbanks. As a starting point, a simple tree-structured filterbank is described for didactical purpose, going through basic concepts and tackling the time-frequency duality in filterbanks. Then, the complex-modulation filterbank solution is introduced, showing both the original and the complex-bandpass structures.

The WOLA filterbank, as integrated in the SignaKlara architecture, is an efficient realization of the complex-modulation filterbank, in its complex-bandpass version. The following sections will describe the structure of the WOLA filterbank and the design of the filters involved in the processing. The filter design is an important aspect of filterbanks.

#### Multirate Filterbanks: the Tree-Structured Decomposition, an Introductory Example

In order to set the initial ideas, consider at first the simplest approach in the filterbank theory, namely the *tree-structured* filterbank. A tree-structured filterbank splits the input time-domain signal into  $M$  sub-band signals, using several cascaded stages of decomposition, as shown on Figure 1 for  $M=4$ .

In the first decomposition stage, the input signal is split in two time-domain signals, which are the respective results of a low-pass and a high-pass filtering process. Both filters have  $F_s/4$  cut-off frequency (*half-band* filters). After filtering, decimation can be applied to those two signals, because their bandwidth is now reduced. Typically, decimation by factor 2 is performed, setting the sampling frequency of both signals to  $F_s' = F_s/2$ . Doing so, and assuming ideal filter characteristics, the sampling frequency is reduced to its minimum. This is the *critically-sampled* situation.

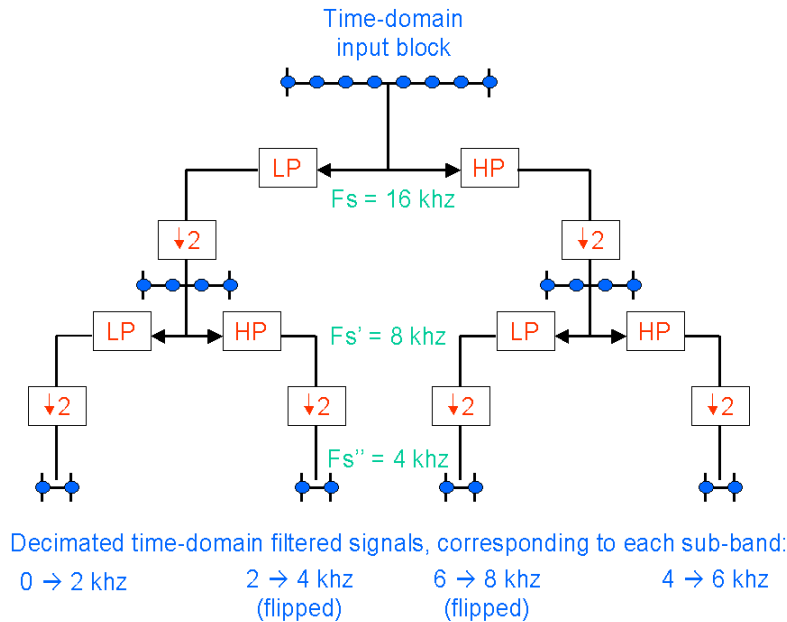
Then, for both sub-bands signals, a new stage of decomposition can be performed, including half-band filtering and decimation. In this way, four sub band are obtained, with sub band signals having sampling

frequencies equal to  $F_s'' = F_s/4$ . Such cascades can be repeated several times, and particular tree structures can be selected, depending on the desired frequency band resolution. Non-uniform filterbanks can be built as well, and used in applications like audio or image coding.

In most applications, a dual tree-structure is used to reconstruct the signal after having performed operations in the sub bands. The reconstruction tree is dual to the decomposition tree. It is made of similar cascaded stages, as shown on Figure 5. Each reconstruction stage performs the addition of two branches containing interpolation (by factor 2) and low-pass or high-pass filtering, respectively. With such a structure, perfect reconstruction can be reached even

using non-ideal filters. In such a case, a special dependency between the decomposition and reconstruction filters is to be satisfied in order that aliasing can be suppressed. Such filters are called *quadrature mirror filters* (QMF, [Vai93]). The decimation/interpolation and filtering processes can be efficiently performed using *polyphase* implementations, saving processing power by a factor of 4.

In practical realizations, blocks of samples are successively processed, just as shown in Figure 1. It is to be noted that neither windowing process nor overlapping between successive input signal blocks is required in a tree-structured QMF filterbank.



**Figure 1. Example of Tree-structured Filterbank Decomposition**

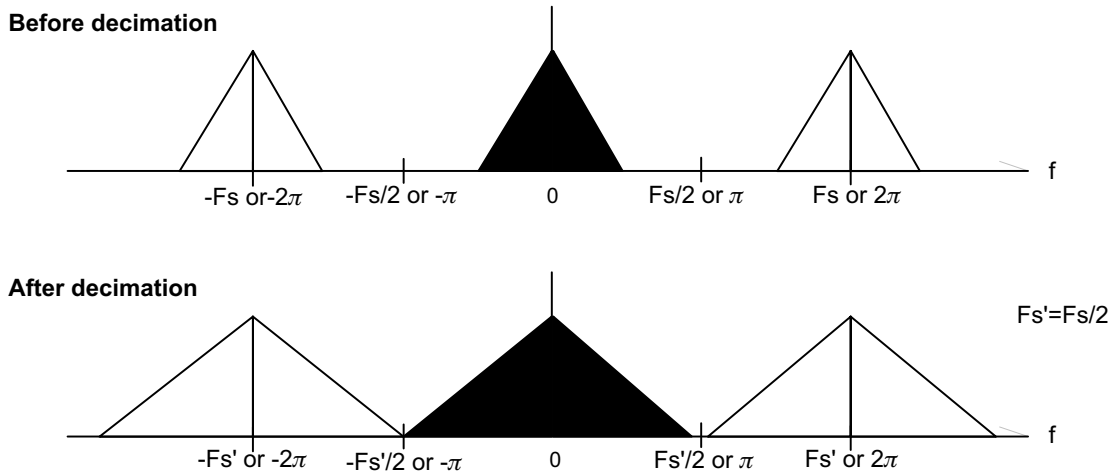
In this particular case, a 16 kHz sampled input signal is considered and successive blocks of eight samples are decomposed into four uniform sub band signals.

Let's shortly review the effects that decimation and interpolation respectively produce on the signal spectrum. Understanding decimation and interpolation is mostly important, in order to correctly interpret the aliasing behavior in filterbanks and chose appropriate related parameters.

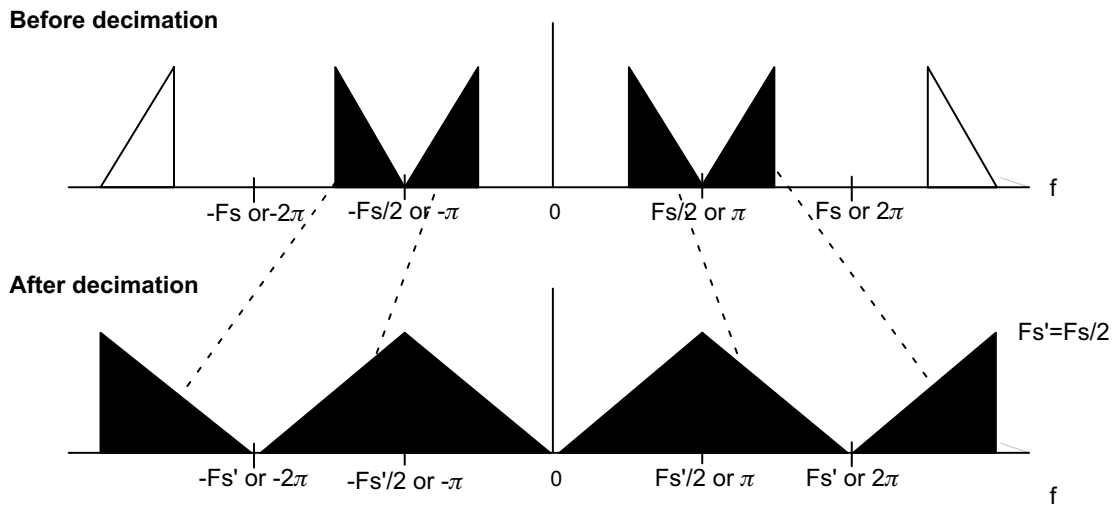
The decimation by factor R of a time-domain signal is actually a "compression in time", which corresponds to an expansion in frequency (time-frequency duality). As a consequence, compared to the original signal spectrum, the frequency spectrum of the decimated signal is expanded by factor R (expansion from the origin of the frequency axis).  $F_s'$ -periodic replications of the expanded spectrum are maintained, as usually for digital signals. Figures 2 and 3

represent what happens when decimation factor is  $R=2$ , for a low-pass and a high-pass spectrum, respectively, using normalized frequency axis ( $f \rightarrow 2\pi f/F_s$  or  $\omega/F_s$  using pulsation  $\omega$ ).

In order to avoid aliasing after decimation, the bandwidth of the original signal should not be larger than  $F_s/2R$ . Hence, after expansion by R, the spectrum bandwidth will not be larger than  $F_s/2$ , still satisfying the Nyquist criterion. Ideal decomposition filters in the tree make sure that such a condition is satisfied. Unfortunately, with real-life non-ideal filters, some aliasing is always present, causing distortions in the signals. Choosing the best filter and filterbank configuration is the challenge, allowing minimizing distortions.



**Figure 2. Original and Expanded Spectra for Decimation Factor R = 2  
Situation for a Low-pass Half-band Spectrum**

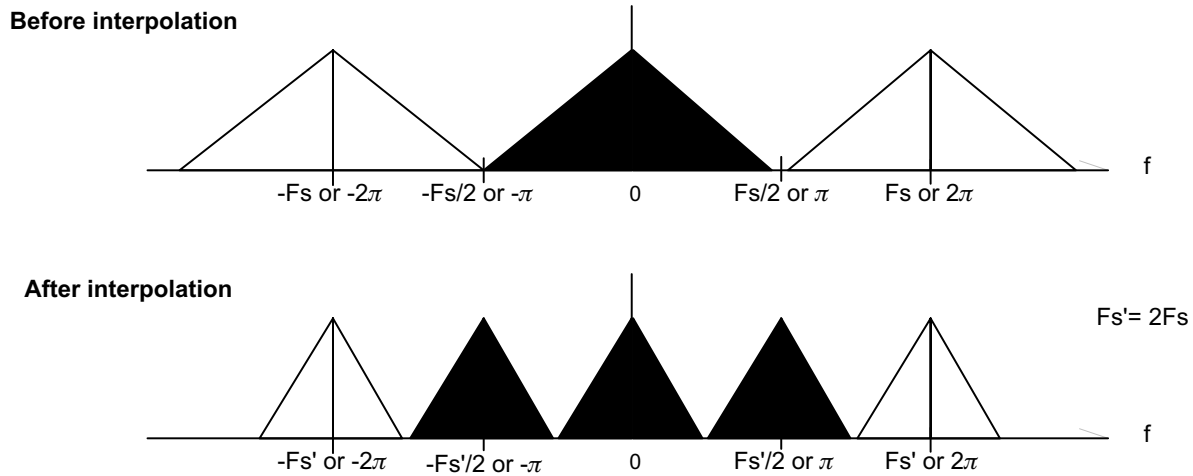


**Figure 3. Input Spectrum and Expanded Spectrum after Decimation  
Situation for a High-pass Half-band Spectrum**

In this case, the resulting base-band spectrum (in black) is a reversed image of the original high-pass spectrum, produced because of  $2\pi$ -periodic replication.

The interpolation by factor R corresponds to an expansion in time. Consequently, the spectrum of the signal gets compressed by factor R (again from the origin of the frequency axis). Because the original spectrum had a periodicity ( $F_s$ ), this periodicity also gets compressed by R, producing images of the original spectrum at interval  $F_s'/R$ . If ideal, the reconstruction filter placed after the

interpolation process totally removes those undesired images. Unfortunately again, using real-life non-ideal filters lets part of the images to stay in the spectrum, producing distortions. Figure 4 represents what happens when interpolation factor is  $R=2$ , showing the images at every  $F_s'/R$  intervals.



**Figure 4. Input Spectrum and Compressed Spectrum after Interpolation**

The resulting images, to be further filtered out, are represented in black.

As an illustration of those properties, coming back to our tree-structured filterbank and assuming perfect half-band filters, the signal spectra met all along the tree-structure are illustrated on Figure 5. Interestingly enough, one can observe that the signals spectra obtained after each “high-pass filtering plus decimation” processes in the decomposition tree get flipped, because of the decimation process (the enlargement of the spectrum makes the symmetric twain part of it fall down below  $F_s/2$ , as shown also in black in Figure 3). As a consequence, after the second decomposition stage, the band order in frequency does not correspond anymore to the band order in the tree.

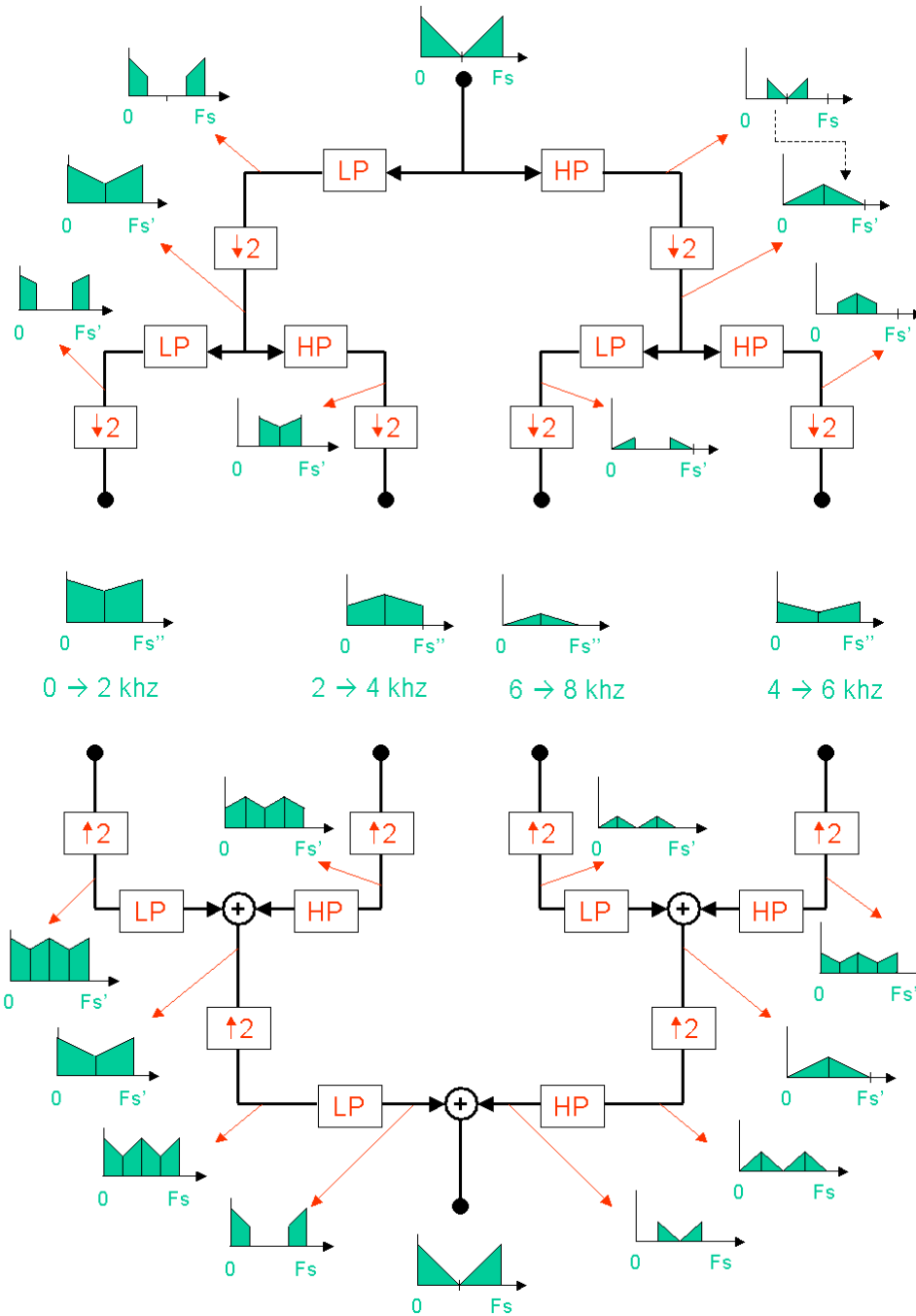
Tree-structured filterbanks have the advantage of being tunable as wished in terms of frequency-band arrangement, but they suffer from two main drawbacks:

- High delay caused by the cascading of several filtering processes. In the case of uniform structures, this problem can be compromised, using only one  $M$ -band QMF decomposition stage instead of  $M$  cascaded 2-band decomposition stages, as done in MPEG-audio filterbanks, notably.
- Aliasing problems in particular applications, due to critical sampling. The consequences of critical

sampling can be observed in : the spectrum images obtained after interpolation are placed against each other, and the use of non-ideal filters will undoubtedly produce aliasing. This aliasing can be cancelled after reconstruction, thanks to the QMF constrained filters. However, this is not true if significant changes are applied to the sub-band signals, before reconstruction.

Those properties make such a critically-sampled tree-structured filterbank appropriate in applications like sub band speech coding, when no delay constraints are required (like in data compression for storage). In such cases, only quantization is applied in the sub bands, and nearly perfect reconstruction is still obtained. Furthermore, critical-sampling reduces the amount of data in each sub band to be quantized to the minimum.

However, in applications requiring significant modifications of the sub bands (like gain application in hearing aid devices, or speech enhancement in communication devices), a different filterbank structure is required, with more flexible configuration possibilities for better management of delay and aliasing properties. The WOLA belongs to the family of such flexible filterbanks.



**Figure 5. Tree-Structured Filterbank: Signal Spectra in Both Decomposition and Reconstruction Trees, for the Configuration Example in Figure 1**

The frequency expansion of the spectrum caused by time-domain decimation can be noticed in the decomposition tree. Similarly, the spectrum compression caused by time-domain interpolation is observed in the reconstruction tree, generating images to be removed by the low-pass or high-pass reconstruction filters.

Before moving to this other filterbank structure, let's consider again the uniform filterbank of Figure 1, realizing that the obtained sub band samples are actually time-domain signals. Effectively, they are just the result of successive filtering and decimation processes. This is actually an important message to remember from this

introductory section: all filterbanks provide time-domain sub band signals, and the WOLA will do so as well. Those M time-domain sub-band signals have spectra in respective sections of the frequency spectrum, corresponding to the particular frequency sub bands. When performing the filterbank process on a block-by-block basis, the output

samples obtained in each sub band can be concatenated to form the filtered audio signal corresponding to this particular sub band. Since the frequency extent of each sub band signal is reduced compared to the original full band signal, the sampling rate of each sub band signal may be reduced through decimation. In the tree-structured filterbank given above for instance, the sampling rate in each sub band corresponds to the original one divided by the global decimation factor of the branch (which is always four in the examples shown in figures).

Finally, let's consider a special case for the input block length. Still considering the uniform filterbank in Figure 1, one could choose a shorter input time-domain block of size 4 instead of 8. In this case, for each block, only 1 output sample instead of 2 would be obtained in each sub band. Since each individual sample represents the frequency content of the original signal for each sub band, one could interpret the  $M=4$  resulting sub band samples as an  $M$ -point time-to-frequency domain transform, being evaluated for each successive block. This is a totally different interpretation. Hence, choosing the appropriate block length, the sub band samples can be interpreted either as time-domain signals or successive frequency spectra. Both interpretations will be considered later in this text. In fact, filterbanks provide both time and frequency domain information, in a dual way. They perform a parallel decomposition into (possibly decimated) time-domain sub band signals. Considering those  $M$  sub band signals at the same time index provides an  $M$ -point frequency domain transform of the input signal.

For now, let's forget about the transform interpretation and keep the time-domain approach in mind.

### Complex-Modulation Filterbanks: Original and Complex-Bandpass Interpretations

Complex-modulation structures use a totally different approach than tree-structured filterbanks and are much more flexible. As the WOLA filterbank belongs to this class, this section allows the reader to understand its principles, characteristics, and parameters. Since, the parameters involved in the following development are the same as those in the WOLA case, there use will be retained. Actually, it is easier to understand the effect of those parameters in the present section, rather than in the WOLA section itself. Effectively, the WOLA is an extremely efficient

implementation of a complex-modulation filterbank and such efficiency tends to make things more difficult to understand, when looking at the block diagram.

### Original Interpretation

In the complex-modulation structure, the time-domain input signal  $x(n)$  is split into  $N$  uniform frequency channels (indexed  $k=0, \dots, N-1$ ), arranged between 0 and  $F_s$ , that is between 0 and  $2\pi$  using normalized frequencies. Each frequency channel is centered on frequency  $\omega_k$ , having bandwidth  $\omega_{\Delta} = 2\pi/N$ . shows the procedure used to decompose (analyze) the input signal  $x(n)$  into the  $k$ -th channel signal  $X_k(m)$ . The same operation is performed for all  $k=0, \dots, N-1$  channels. illustrates the reconstruction (synthesis) steps, reconstituting signal  $\hat{x}(n)$  from the  $N$  sub-band components  $\hat{X}_k(m)$ . As in the usual filterbank

approach, let's consider  $X_k(m)$  as a time-domain sequence in channel  $k$ ,  $m$  being the index of the time-domain samples.  $\hat{X}_k(m)$  is a processed version of  $X_k(m)$ .

The terminology "complex-modulation" comes from the process performed in step 1 (see Figure 6). In fact, the time-domain input signal  $x(n)$  is multiplied by the complex exponential  $\exp(-j\omega_k n)$ . In the frequency domain, this multiplication corresponds to a frequency shift of the full spectrum by  $-\omega_k$ , positioning channel  $k$  at the origin of the frequency axis. Examining this operation in more detail, one realizes that the resulting shifted spectrum is asymmetric since the point of symmetry is no longer at  $f=0$ , as shown on Figure 8. As a consequence, in the time domain, the signal becomes complex (real signals always feature symmetric spectra around  $f=0$ , while complex signals do not).

Then, as a second step, a low-pass filter  $h(n)$  with cut-off frequency equal to  $\omega_{\Delta}/2 = \pi/N$  is applied to the shifted spectrum, isolating channel  $k$  (now centered at the origin) by removing the  $N-1$  other channels. This filter is the analysis filter. Finally, in Step 3, decimation by factor  $R$  of the filtered signal is performed, and the signal  $X_k(m)$  representing channel  $k$  is obtained, still as a complex signal since its spectrum is not symmetric around  $f=0$ . This is an important difference compared to tree-structured filterbank, for which real valued sub bands signals (their spectra are symmetric, see Figure 5) were obtained after filtering and decimation of the real time-domain input samples.

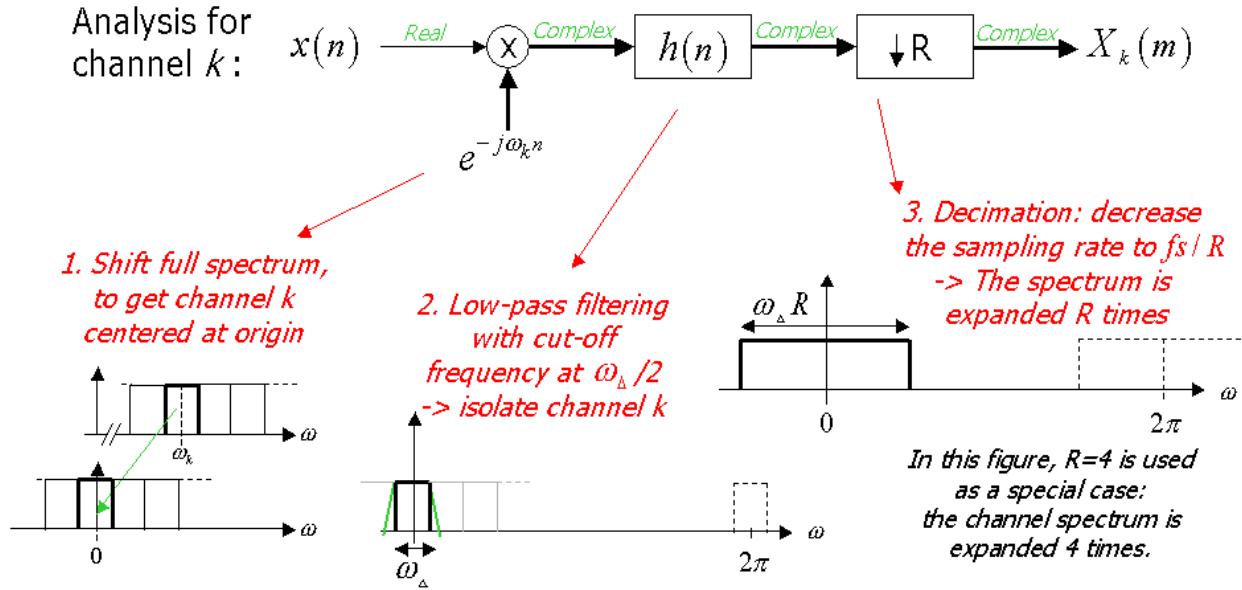


Figure 6. Complex-modulation Filterbank: Analysis Steps for one Particular Channel  $k$

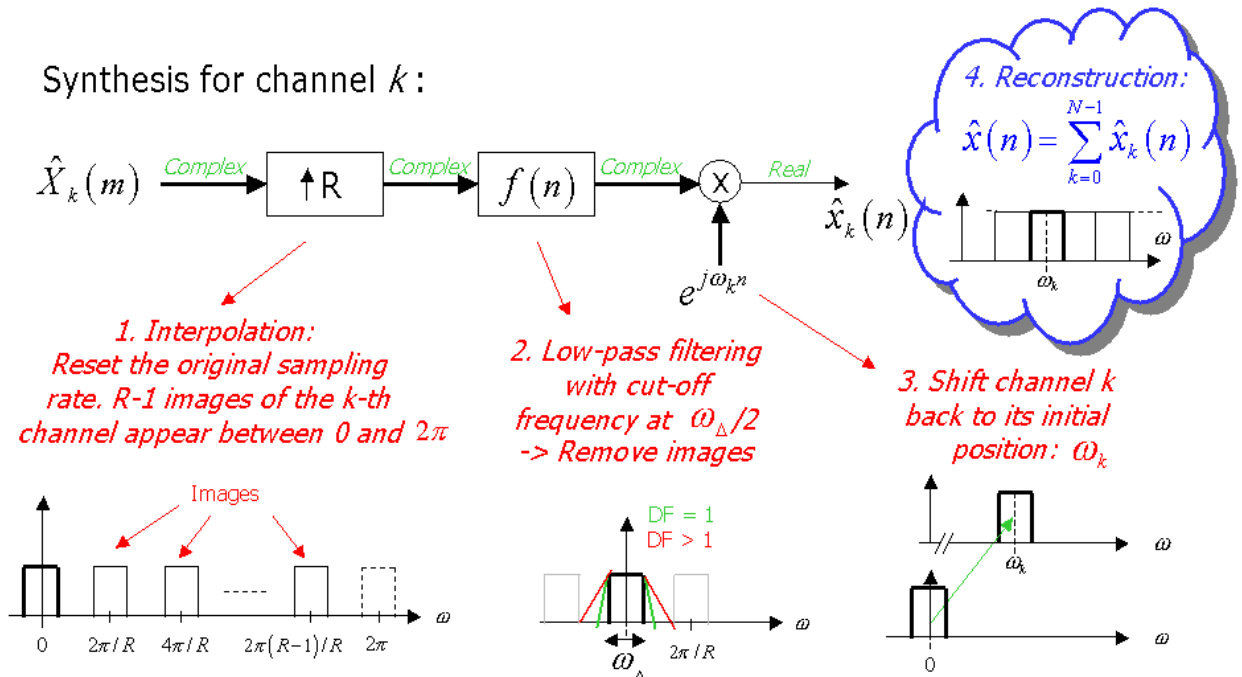


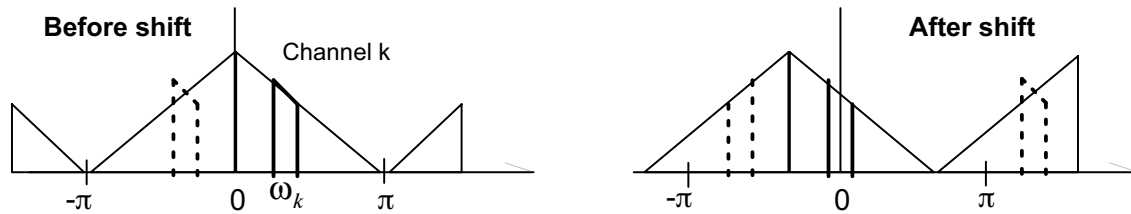
Figure 7. Complex-Modulation Filterbank: Synthesis Steps for one Particular Channel  $k$

Another difference to be mentioned, comparing both filterbank structures, is the number of resulting sub-bands or channels, after decomposition. In the previous section, the  $M$  sub bands resulting from a uniform tree-structured decomposition were distributed between 0 and the Nyquist frequency  $F_s/2$ . No information about frequencies higher than  $F_s/2$  (that is  $\pi$ ) was provided. Assuming real input signals, this was not a problem because of the spectrum symmetry. Conversely, complex-modulation filterbanks

provide  $N$  channels, covering the full frequency range from 0 to  $F_s$  (or  $2\pi$ ). This means that complex input data could be processed as well. The WOLA has a stereo mode that takes advantage of this property by packing two audio input channels together and treating it as complex data. For clarity and simplicity however, we will assume only real values are used, and the second part of the spectrum is the complex-conjugate of the first half. In this text, the following terminology convention is adopted:  $N$  channels



are considered between 0 and  $F_s$  (or between 0 and  $2\pi$  using normalized frequencies), while  $N/2$  bands or sub bands are considered in the range  $[0, F_s/2]$ .



**Figure 8. The Complex-Modulation (Multiplication of the Time-Domain Signal with  $\exp(-j\omega_k n)$  is Equivalent to a Shift of the Frequency Spectrum Consequence: No Spectrum Symmetry at  $f = 0$  is Complex Signal in Time**

At reconstruction (Figure 7), the steps are performed in reverse. For each channel  $k$ ,  $\hat{X}_k(m)$  is first interpolated, producing images at intervals  $2\pi/R$ . Then, a low-pass filter  $f(n)$ , the synthesis filter, is applied in order to remove those images. The passband to be preserved is  $\omega_{\Delta}/2$  which is  $\pi/N$ . The first image occurs at frequency  $2\pi/R - \pi/N$ . When the decimation factor  $R$  is smaller than the number of channels  $N$  (an oversampled situation), the specifications for this reconstruction filter are somewhat relaxed because of the gap between the required passband and the first image (see Figure 7). The synthesis filter may be made identical to the analysis filter and indeed this choice minimizes the distortion due to imaging. However, in many filterbank configurations, adequate image rejection is possible with much lower delay taking advantage of the relaxed specifications choosing its order to be much lower. In such cases, and if using FIR filters, the impulse response of the synthesis filter could be shortened by simply using a decimated version (by DF) of the analysis filter impulse response. It should be noted that DF is merely a notational convenience and has no link with the decimation factor  $R$  of the filterbank. As the WOLA implementation will only support FIR filters, the following lines will assume FIR filters. Finally, as a third step, the channel is shifted back to its original position  $\omega_k$ , using a complex modulation. After performing those three steps for every channel, the re-synthesized signal is obtained, adding all  $N$  components. Using examples, the Annex 1 of this document illustrates the whole analysis and synthesis process of the complex-modulated filterbank. The spectra are shown at each step, allowing to better understand what happens to the signal going through the filterbank.

Compared to the tree-structured solution, the complex-modulation filterbank is much more flexible. In fact, the decimation factor  $R$  can be selected as desired in the range  $R=1$  (undecimated) to  $R=N$  (critically-sampled), independently of the number of frequency channels  $N$ . As a consequence, the amount of aliasing can be better controlled and compromised with the filter sharpness. A compromise between aliasing rejection in the frequency domain (channels) and imaging rejection in the time-domain (after

reconstruction) is possible. Basically, aliasing is produced in two ways:

- Aliasing at analysis is caused by the non-ideal characteristics of  $h(n)$ . Some components of all the other-channels stay present in the  $k$ -th channel, because of lack of filter sharpness and non-ideal stop-band profile (ripples). When decimation is performed, those remaining components from all other channels are aliased in the  $k$ -th channel.
- Imaging at synthesis is caused by the non-ideal characteristics of  $f(n)$ . As a consequence, some parts of the interpolation images of the  $k$ -th channel persist, and will be added to all other channels because of lack of sharpness and significant ripples. Among the remaining images of the  $k$ -th channel, the ones near channel  $k$  have a higher level.

Considering the paragraphs above, and looking at Figures 6 and 7, the following major parameters can be identified in the complex-modulation filterbank:

- Number of channels  $N$  (or number of frequency bands  $N/2$ )
- Decimation factor  $R$  in the filterbank
- Analysis filter  $h(n)$  of length  $L_a$
- Synthesis filter  $f(n)$  of length  $L_s (= L_a/DF)$

The choice, and influence of those parameters is the key point in the filterbank design, having consequences on the amount of aliasing, the group delay, the calculation load and, of course, the frequency band resolution. As ideal filters only exist on paper, trade-offs will have to be performed in order to keep the unavoidable aliasing inaudible. All those parameters will be discussed in details later in this text, based on examples. As a first contact, it can be observed from Figures 6 and 7 that:

- Increasing the number of channels  $N$  means reducing the widths  $\omega_{\Delta} = 2\pi/N$  of the channels. As a consequence, if the same amount of channel crossover is to be maintained, sharper (that is longer) filters are required. This is particularly important when different processing is to be applied in adjacent channels. The analysis filter is especially concerned, even if a sharper

synthesis filter will also contribute to improve quality, by further removing some remaining out-of channel components.

- Increasing the decimation factor  $R$ , while keeping the same number of channels  $N$ , moves the interpolated images closer to each other (Figure 7). As a consequence, the synthesis filter must be sharper (longer) in order to keep the same level of image rejection. Usually, an additional parameter is defined to represent the closeness of the interpolation images in Figure 7. This parameter is called the oversampling factor (OS), and is defined as  $OS = N/R$ . Increasing  $N$ , when OS is not changed (that is also increasing the decimation factor  $R$ ) maintains the same separation between the interpolation images, and consequently, the same aliasing properties at synthesis. It should be noted that increasing OS increases the calculation load.
- Increasing the length  $L_a$  of the analysis filter, or the length  $L_s$  of the synthesis filter increases the quality, but also the delay in the filterbank. Effectively, the delay associated for a symmetric FIR filter of length  $L$  is  $L/2$ . The calculation load is also increased.
- As noted above when OS is greater than 1 (that is  $N > R$ ), decreasing the length  $L_s$  of the synthesis filter compared to the analysis filter is possible. This reduces both group delay and calculation load but decreases the image rejection capability. Using a synthesis filter of length  $L_s = L_a / DF$  is generally possible, applying the following rule of thumb:  $DF$  should be lower or equal to  $OS$ .

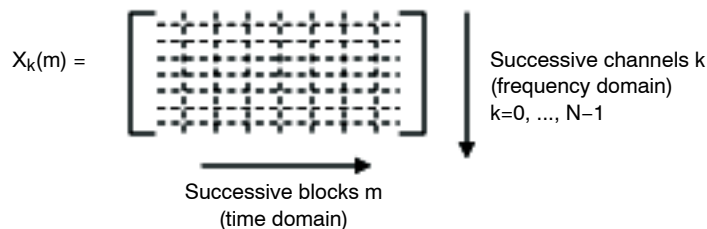
**Complex-Bandpass Interpretation**

In the structures shown above, at the beginning of the analysis, the fullband spectrum was shifted by  $-\omega_k$ , in order to center the  $k$ -th channel at the origin of the frequency axis. In this way, the same unique low-pass filter  $h(n)$  may be used for all successive channels to be processed. Conversely, in the complex-bandpass interpretation, the fullband spectrum is not shifted. Instead, separate bandpass filters,  $h_k(n)$ , are used to isolate the other  $k$  channels. As shown in Figure 10, this filter is actually a frequency shifted/modulated (multiplied by  $\exp(j\omega_k n)$ ) version of the low-pass *prototype* filter  $h(n)$ . Because of the complex-modulation applied on its impulse response, the bandpass filters  $h_k(n)$  have complex coefficients.

In the complex-bandpass structure, the three following stages are performed when analyzing channel  $k$ . At first, the input signal is filtered by  $h_k(n)$ . Then, the obtained complex-bandpass signal is decimated by factor  $R$ . Finally, its spectrum is shifted to the origin, using a complex-modulation. Hence, this time, the complex-modulation by  $\exp(-j\omega_k n)$  is applied to the individual channels, after decimation. At synthesis, the opposite steps are performed, as shown on Figure 11: the channel is first shifted back to its original frequency position, then interpolation is applied, producing images, and finally, those images are filtered out using the synthesis filter  $f_k(n)$ . After performing those steps for every channel, the re-synthesized signal is obtained, adding all  $N$  components. Using examples, the Annex 2 of this document illustrates the whole analysis and synthesis process of the complex-modulated filterbank, considered in the complex-bandpass structure. Again, the spectra are shown at each step, allowing to better understand what happens to the signal going through the filterbank.

The complex-bandpass structure is equivalent to the original complex-modulation scheme, and the influence of the involved parameters is the same. However, the operations sequence is more advantageous compared to the original complex-modulation scheme. Sometimes, the last analysis step and the first synthesis step (the opposite complex-modulations) can be avoided. Effectively, those complex-modulations only have influence on the phase of the signal (details are available in the section dedicated to overlap-add configurations). As a consequence, if the process to be performed in the frequency domain only concerns the amplitude, it is no use to perform those two opposite complex-modulations.

Finally, just like in the tree-structured filterbank, for each channel  $k$ , the analyzed signal  $X_k(m)$  can be interpreted as a decimated time-domain sequence (complex),  $m$  being the sample index. Alternatively, whenever only one sample is obtained at a time in each channel  $k$ ,  $X_k(m)$  becomes the  $N$ -point spectrum for the  $m$ th block of input signal. The dual interpretation can be illustrated, looking at matrix  $X_k(m)$ , where each line represents the time-domain signal in channel  $k$ , while each column represents the spectrum for block  $m$ :



**Figure 9. Dual Time and Frequency Interpretation of the Channel Signals  $X_k(m)$**

Practically, in all usual implementations like the one described in the next section, the length of the time-domain input block is always chosen in a way to obtain only one sample at a time in each frequency channel. This reduces the blocking delay to the minimum. Therefore, the length of each successive input block is  $R$ , the decimation factor.

**Weighted-Overlap-Add Implementation: the WOLA**

Two methods are described in the literature, for efficiently implementing complex-modulation filterbanks. The first one uses polyphase structures. This method is particularly appropriate in critically-sampled cases. However, its use becomes more difficult in situations where the oversampling ratio should be fully flexible. In such a case, a second method, the weighted overlap-add (WOLA) implementation is recommended. For both techniques, some part of the processing required during analysis can be expressed in terms of an  $N$ -point *discrete fourier transform* (DFT). Then, implementing the DFT using an FFT accelerates the calculation in this part of the processing. The same situation occurs in the synthesis process where a portion of the calculation may be expressed in terms of an *inverse discrete fourier transform* (IDFT). For that reason, such filterbanks are sometimes called *DFT filterbanks*. However, it is important to realize that such DFT filterbanks are really complex-modulation filterbanks, with all their flexibility. No comparison stands with a simple  $N$ -point FFT, in which the frequency resolution is directly linked to the time-domain input window. Using a DFT filterbank like the WOLA, time and frequency resolutions can be selected independently. Confusions between using simple  $N$ -point

FFT's and DFT filterbanks such the WOLA, can easily occur, because the FFT is used as the final step of the filterbank analysis, providing complex-conjugate samples as filterbank output, just like an FFT applied to  $N$  input samples. However, and this is the most important message of this tutorial, the WOLA filterbank is not an FFT. The WOLA is an efficient implementation of a complex-modulation filterbank, based on its complex-bandpass interpretation.

As mentioned before, in the WOLA, the filterbank process is performed on a block-by-block basis, choosing the input block size to be equal to the decimation factor  $R$ . As a consequence, only one sample at a time is obtained in each channel, and the filterbank process can also be considered as a time-to-frequency transform, providing a spectrum at each new block. As a main efficiency aspect, and converse to the complex-modulation way of thinking, the WOLA implementation processes all channels simultaneously, sharing part of the calculation among the channels. In Figures 12 and 13, the block diagrams of the WOLA analysis and synthesis are shown and some qualitative explanations are given below. Deriving the complete equations of the WOLA implementation is out of scope for this tutorial, and understanding the mathematical steps ([Cro83]) leading to the WOLA structures of Figures 12 and 13 is not necessary. In this text, the reader is simply invited to consider the WOLA as a fast and efficient realization of the complex-modulation filterbank, thinking about the parameters ( $N$ ,  $L_a$ ,  $L_s$ ,  $R$ , or  $OS$ ), as stated in that scheme.

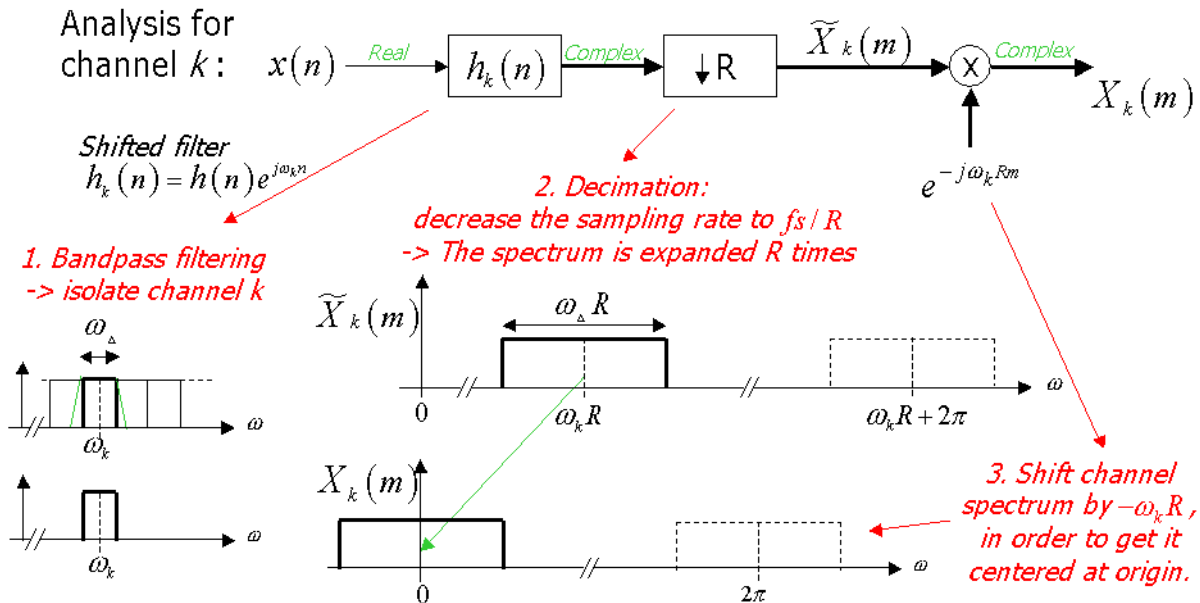


Figure 10. Complex-modulation Filterbank: Analysis Steps for one Particular Channel  $k$

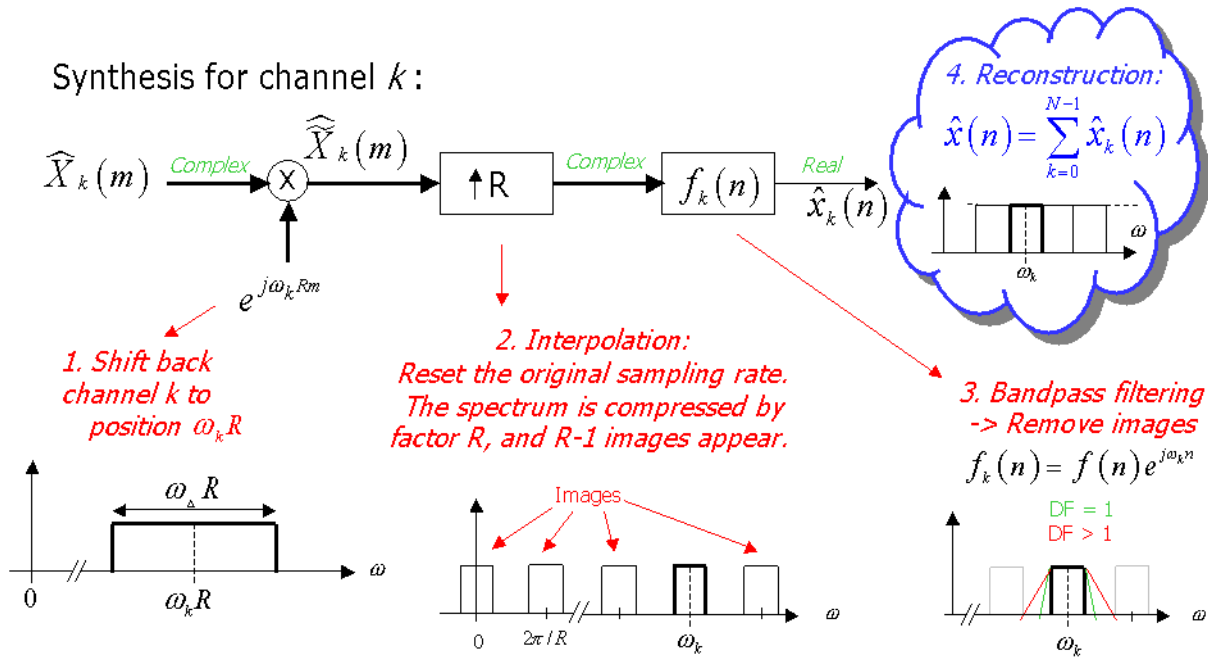


Figure 11. Complex-Modulation Filterbank: Synthesis Steps for one Particular Channel  $k$

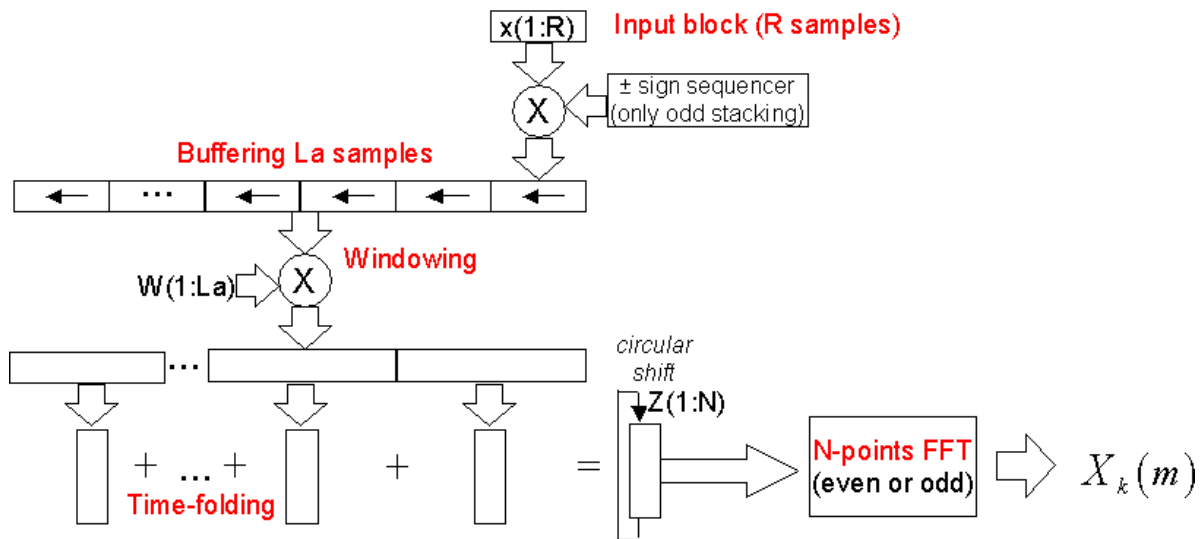


Figure 12. WOLA Filterbank: Block Diagram for Analysis

# AND8382/D

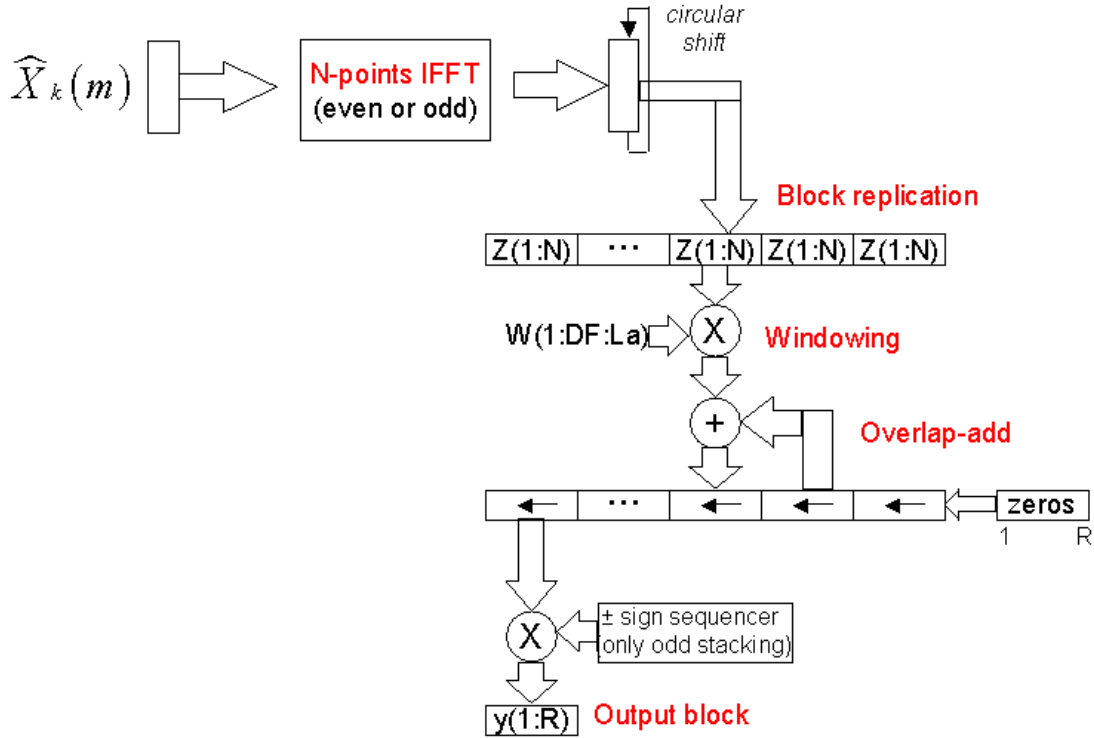
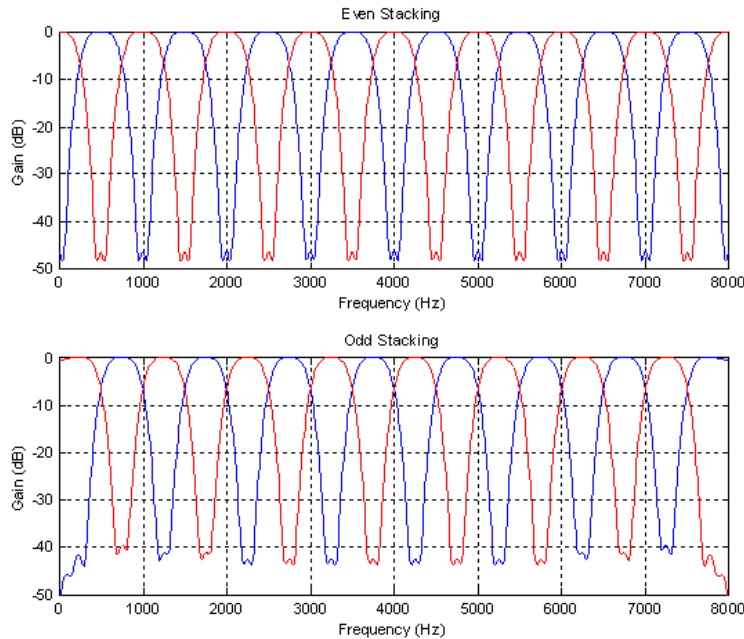


Figure 13. WOLA Filterbank: Block Diagram for Synthesis

As an additional parameter, the WOLA implementation allows for two different channel stacking modes: the usual even stacking mode with channels centered at  $\omega_k = 2\pi k/N$  ( $k=0, \dots, N-1$ ) and the odd-stacking mode, with channels

centered at  $\omega_k = 2\pi(k+0.5)/N$ , which is sometimes preferred. Figure 14 shows the respective band positions for both modes.



**Even Stacking:**  
 $N/2+1$  bands, with a DC and a Nyquist band having half bandwidths

**Odd Stacking:**  
 $N/2$  bands, uniformly distributed between 0 and  $fs/2$

Figure 14. Even and Odd Stacking Modes in the WOLA Filterbank: Example for 16 Bands,  $fs = 16$  kHz

In the WOLA analysis scheme, the filters are applied as a windowing ( $W$  is the prototype filter  $h(n)$  in Figures 12 and 13 above), weighting the successive input blocks. Then, a time-folding operation is performed in order to get only one block of length  $N$ , and to be able to calculate its FFT. Finally, the complex-modulation (Step 3 in the complex-bandpass structure of Figure 10) is performed before the FFT, as a time-domain circular shift. More details about the interpretation of this circular shift are discussed in a later section. The synthesis steps correspond to the reciprocal process (inverse FFT, opposite circular shift, block replication, synthesis windowing, and overlap-add). A few additional processing steps are required for the odd stacking mode (sign sequencer).

In ON Semiconductor's WOLA hardware realization, some constraints are imposed to the filterbank parameters. The filter length  $L$ , input block size  $R$ , and number of channels  $N$  have to be powers of two. Furthermore, their maximum is 256. Minimum values are 32 for the filter lengths, while the input block size can be as low as 2.  $DF$  also has to be a power of 2. Practically, the number of channels  $N$  should always be higher than  $R$ , and equal or lower than the filter lengths. The most usual filterbank configurations are available as pre-programmed "microcodes", and included in ON Semiconductor's evaluation and development kit. Other configurations can be designed and provided to customers for individual needs. In some cases, special microcodes can be provided to customers for configurations going beyond the parameter limits (using for example  $N = 512$ ). Contacting ON Semiconductor's support is recommended when there are needs for special configurations.

In ON Semiconductor's DSP solutions, the WOLA filterbank co-processor is habitually used in the following scheme. It should be noted that the hardware only provides the bands, ( $N/2$  values in odd stacking, respectively  $N/2+1$  in even stacking), while the complex-conjugated channels are ignored because real input signal is assumed. A

processing mode is also provided which provides  $N$  bands when the input is complex (stereo).

- **WOLA Analysis:** Filterbank analysis applied to every new block of  $R$  input audio samples, providing one Sample  $X_k(m)$  in every band  $k$ .
- **WOLA Gain Application:** Multiplication of the  $X_k(m)$  by real or complex gains  $G(k)$ .
- **WOLA Synthesis:** Reconstruction of a processed audio signal.

In terms of number of samples, the algorithmic delay in the filterbank could be expected to be  $La/2 + Ls/2$ , corresponding to both filters delay. Then, the total group delay, involving also the block buffering delays at both analysis and synthesis would be  $La/2 + Ls/2 + 2R$ . However, let's have a closer look to this delay calculation thinking about it in terms of a particular sample being processed. This sample is first buffered in the input block (FIFO), waiting for  $R$  samples. Then it enters the filtering process, going out of it  $La/2$  samples later because of the analysis filter delay. This makes already a delay of  $R + La/2$  samples. And now, there is a trick: what happens? Immediately after analysis, *within the same block or processing loop*, this sample enters the synthesis filtering operation, which is started immediately after the analysis. This is like a "going back in time" operation by  $R$  samples. As a consequence, one block of  $R$  samples will be spared when counting the total amount of delay. Our sample will then come out of the synthesis  $Ls/2$  samples later, that is with a delay of  $(R + La/2) - R + Ls/2$  in respect to the time it entered the input FIFO. As a last step, the sample will still have to wait  $R$  samples in the output buffer (FIFO), and the total delay finally reaches  $La/2 + Ls/2 + R$  samples. This process is illustrated in . As a conclusion, the algorithmic delay (which can be observed in Matlab) is  $La/2 + Ls/2 - R$ , and the total group delay, including the blocking operations is  $(La/2 + Ls/2 - R) + 2R = La/2 + Ls/2 + R$ .

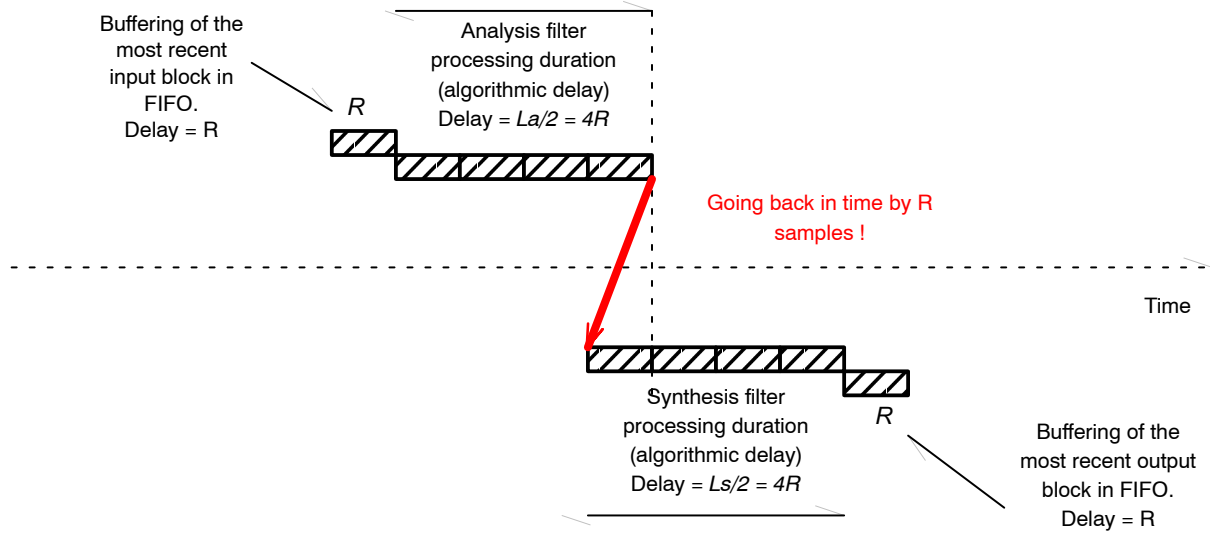


Figure 15. Calculation of the Group Delay, for a Configuration where  $R = La/8 = Ls/8$

As a final observation, it should be noted that the WOLA is a uniform filterbank. Non-uniform filterbank structures cannot be realized directly, using such an efficient complex-modulation structure. However, depending on delay and complexity constraints,  $N$  can be increased, in order to reach the finest required frequency resolution in the non-uniform filterbank (generally in the lower part of the spectrum). Then, the higher frequency bands can be grouped together appropriately, saving processing. In such cases, the counterpart is of course the delay, which is increased because of the higher number of bands. Effectively, generating a higher number of bands needs longer filters for reducing the aliasing. Only the narrowest frequency band required fundamentally determines time delay through a filterbank. It makes no difference to the filterbank delay whether a non-uniform filterbank is realized directly or is realized by grouping bands together from a highly efficient uniform filterbank.

**Prototype Filter Design**

Both analysis and synthesis low-pass prototype filters should be designed carefully for appropriate aliasing and imaging rejection. Using the WOLA realization, only FIR filters can be used because the impulse responses must have finite durations as required in the time-folding operation. The FIR filters can be designed with any usual method. For better understanding, this section deals with considerations about filter design aspects, considering the *window design* method for linear-phase FIR filters. In the window design method, the following steps are performed, while designing a low-pass filter prototype for the filterbank:

- Design of an ideal rectangular low-pass characteristic in the frequency domain, with cut-off frequency at  $\omega_c = \omega_A/2 = \pi/N$ .
- Calculation of the corresponding impulse response, which is a sinc function (infinite length), crossing the

zero-axis every  $N$ -th samples, starting from the main lobe:

$$h_{in}(n) = \text{sinc}(n\omega_c/\pi) = \text{sinc}(n/N)$$

- Truncation of the sinc function, in order to get a finite-duration impulse response of length  $La$  (for the analysis filter).

$$h_{trunc}(n) = \text{sinc}(n\omega_c/\pi) \text{ for } n \text{ in } [-La/2 \rightarrow La/2-1]$$

$$h_{trunc}(n) = 0 \text{ otherwise}$$

- Time-shifting of the truncated response by  $La/2$  samples, in order to get causal behavior (introduces a delay equal to  $La/2$ ).

$$h_o(n) = \text{sinc}((n-La/2)\omega_c/\pi), \text{ for } n \text{ in } [0 \rightarrow La-1]$$

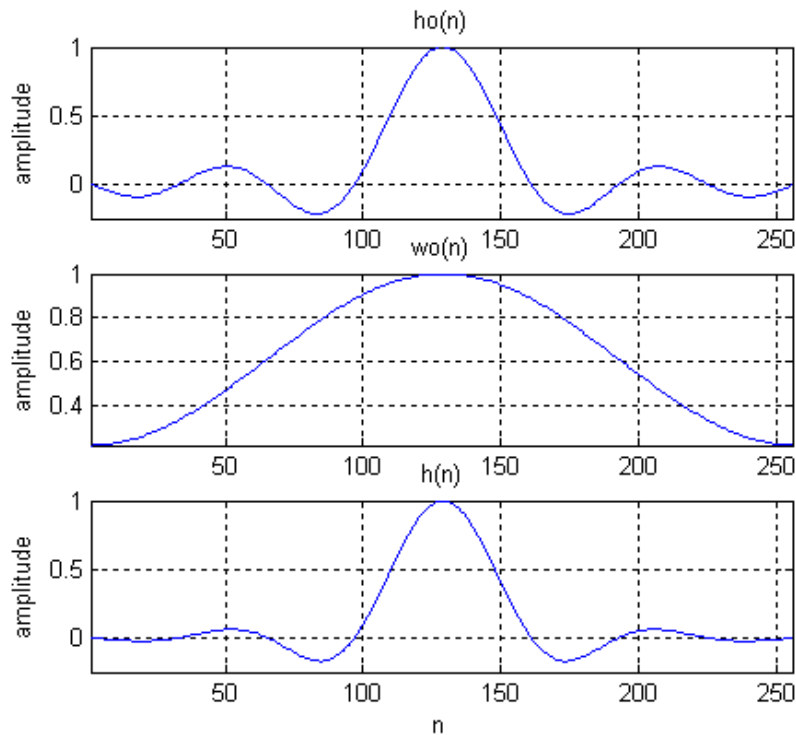
$$h_o(n) = 0 \text{ otherwise}$$

- Windowing of the truncated impulse response (sinc), in order to reduce Gibbs Phenomenon, and get some specified frequency response characteristics (passband and stopband ripples, transition bandwidth). Noting this window  $w_o(n)$ , the final impulse response is:

$$h(n) = w_o(n) h_o(n) \text{ for } n \text{ in } [-La/2 \rightarrow La/2-1]$$

Hence, in such a design, the filter impulse response is built by multiplying the sinc function  $h_o(n)$  with a window  $w_o(n)$ . This is shown in Figure 16.





**Figure 16. Design of a Filter Impulse Response for  $L_a = 256$  and  $N = 32$  (the Cut-Off Frequency of the Prototype Filter is Therefore  $p/32$ , Which Sets the Shape of the sinc)  $w_o(n)$  is the Brennan Window**

The sinc function holds information about the cut-off frequency, while the window controls the transition bandwidth as well as the passband and stopband behavior. After setting a cut-off frequency, the window totally drives the way the aliasing is produced. As a consequence its choice is crucial, and the reader is referred to the dedicated filterbank literature for more details [Cro83, Vai93]. Many types of windows are available (Hamming, Blackmann, Kaiser, ...), each one having specific characteristics and advantages. In order to facilitate this window design, ON Semiconductor has extensively studied the problem in the filterbank framework and provides default windows or filters, which are appropriate in most configurations. As a consequence, using those proposed filters is highly recommended, while customization is always possible. The default filters are always used by the microcodes. When necessary, a customized filter can be written in the appropriate DSP memory location, for use by the WOLA. Then it replaces the default window.

As an important terminology issue, it must be clear that the WOLA windowing process, as illustrated in both Figures 12 and 13 actually involves the *impulse response* of the filterbank analysis and synthesis filters,  $h(n)$  and  $f(n)$  respectively, and not just the windows  $w_o(n)$  used in the filter design. As a consequence,  $W$  is  $h(n)$  or  $f(n)$  in those figures. Because of the particular WOLA structure, the filtering operations are not performed as convolutions, but *appear* as simple multiplications of the time-domain audio samples with the impulse response. In fact, considered together with the other WOLA blocks, those windowing operations really

participate to the filtering. As a consequence, in a generic approach, they should not be interpreted as the windowing operations performed prior to an FFT (for preventing block limit effects), even if both interpretations would match in some particular configurations (more details later). Once again, this window  $W$  is the filter impulse response, that is the product of the sinc with the window used in filter design.

#### The Brennan Window

The algorithmic delay coming from the filterbank depends on both analysis and synthesis filter lengths. As a consequence, it is interesting to try and reduce those lengths, provided the amount of aliasing does not increase significantly. Because of the complex-modulation filterbank properties, it is often possible to choose a synthesis filter, which has a shorter length compared to the analysis filter. When the oversampling ratio is high enough, such a solution has reduced impact on aliasing. As a consequence, using parameter DF, the synthesis filter could be a decimated version of the analysis filter, which additionally allows storing only one set of filter coefficients in DSP memory.

The Brennan window is the default window  $w_o(n)$  proposed by ON Semiconductor to build the filters  $h(n)$  and  $f(n)$ , multiplying the sinc expression, as presented above. Like the Hamming window, the Brennan one is issued from the general cosine function framework  $\alpha - \beta \cos(2\pi n / L_a)$ , setting  $\alpha = 0.61$ , and  $\beta = 0.39$ :

$$\text{Brennan}(n) = 0.61 - 0.39 * \cos(2\pi n / L_a)$$

for  $n$  in  $[0 \rightarrow L_a - 1]$



This window is always used as the default in the microcodes, except if  $N=La=Ls$ , which is a special kind of configuration called *overlap-add*, discussed in more details later. The Brennan window was designed for all situations in which  $DF$  is higher than 1. Despite of that, it is important to note that in the hardware, this window is also used when  $DF = 1$  (provided that the configuration is not an overlap-add one). However, in such cases, better filters could improve the frequency response in the passband. Explanations about the design of such filters are given below, as well as in Chapter 3 (Configuration 8). Let's have a look to the frequency response of the filters, when the analysis and synthesis impulse responses are generated with the Brennan window. Respectively:

$$h(n) = \text{Brennan}(n) \text{ sinc}((n-La/2)\omega_c/\pi)$$

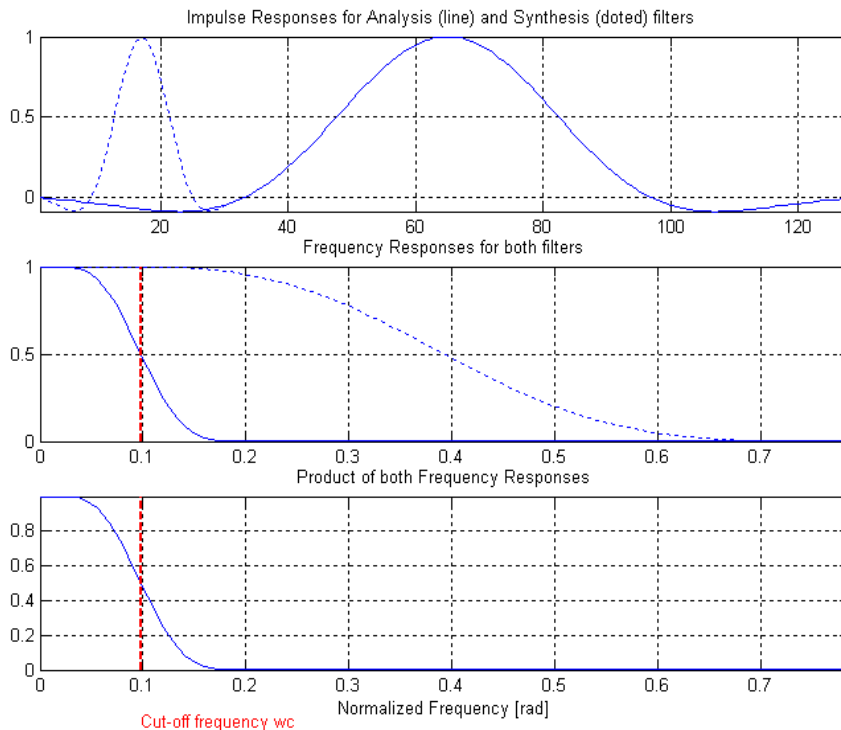
for  $n$  in  $[0 \rightarrow La-1]$

$$f(n) = h(DF n)$$

for  $n$  in  $[0 \rightarrow Lsa-1]$

Those impulse responses are shown in the first plot of Figure 17, assuming a filterbank configuration including

$La = 128$ ,  $DF = 4$ , and  $N = 32$  channels. One can observe that the synthesis filter is four times shorter than the analysis filter. As a consequence, in the frequency domain, the frequency response of the synthesis filter is expected to be four times expanded compared to the analysis filter one, which can be seen on the second plot of Figure 17, as well as the location of the channel cut-off frequency  $\omega_c = \pi/N$ . At the cut-off frequency, the analysis filter frequency response has a value of 0.5, which corresponds to an attenuation by 6 dB. On the opposite, the synthesis filter value is nearly 1. As a result, when performing an analysis, followed by a synthesis, the global attenuation at the cut-off frequency, caused by one particular band is approximately  $0.5 * 1 = 0.5$ , as expressed in the third plot of Figure 17. Finally, considering that the same attenuation amount is generated by the subsequent band at the cut-off frequency, the original level (that is 1) is restored when adding both channels (which is done at Step 4 in filterbank synthesis). Of course, some ripples are present, notably because the synthesis filter is not exactly 1 at the cut-off frequency.



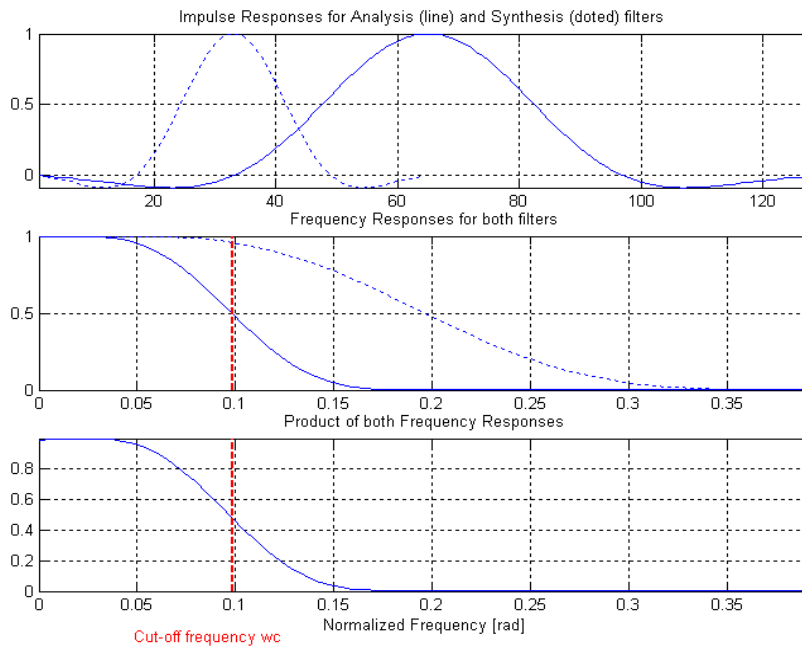
**Figure 17. Impulse Responses (first plot) and Frequency Responses (Second Plot) of Both Analysis and Synthesis Filters, in the Case  $La = 128$ ,  $Ls = La/DF = 32$ ,  $N = 32$**

The third plot shows the product of both frequency responses.

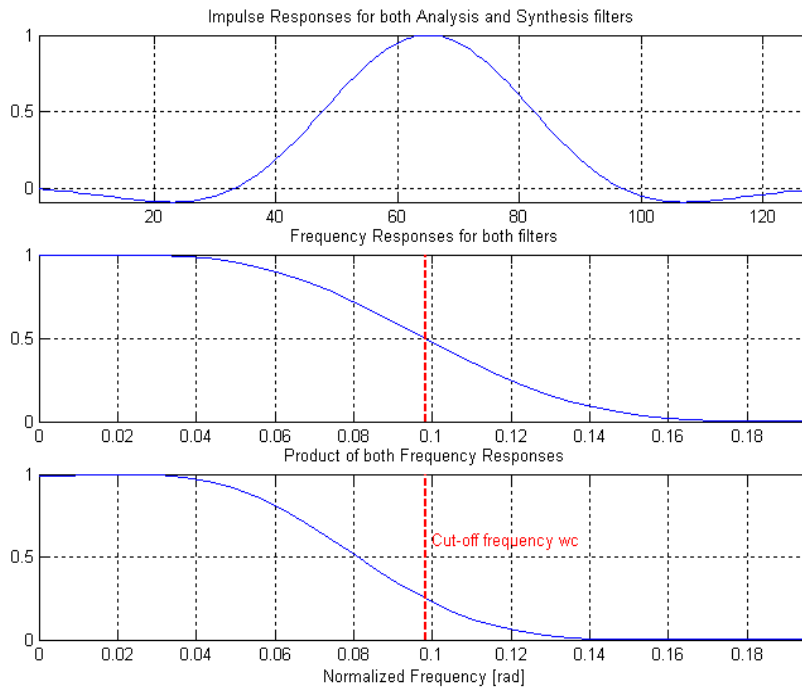
Let's now change the configuration, decreasing  $DF$  to 2 (Figure 18) In such a case, the synthesis filter frequency response is sharper, but its value at the cut-off frequency is still close to 1 as can be observed looking at the second plot. As a result, the global analysis-plus-synthesis response is again equal to 1, even if more ripples can be expected, compared to  $DF=4$ .

As a final experiment, let's keep the synthesis filter identical to the analysis one, setting  $DF = 1$ . In this case, it appears on Figure 19, that the frequency response of  $f(n)$  is not anymore equal to 1 at the cut-off frequency. As a consequence, when performing the analysis followed by the synthesis, the product of both responses is 0.25 instead of

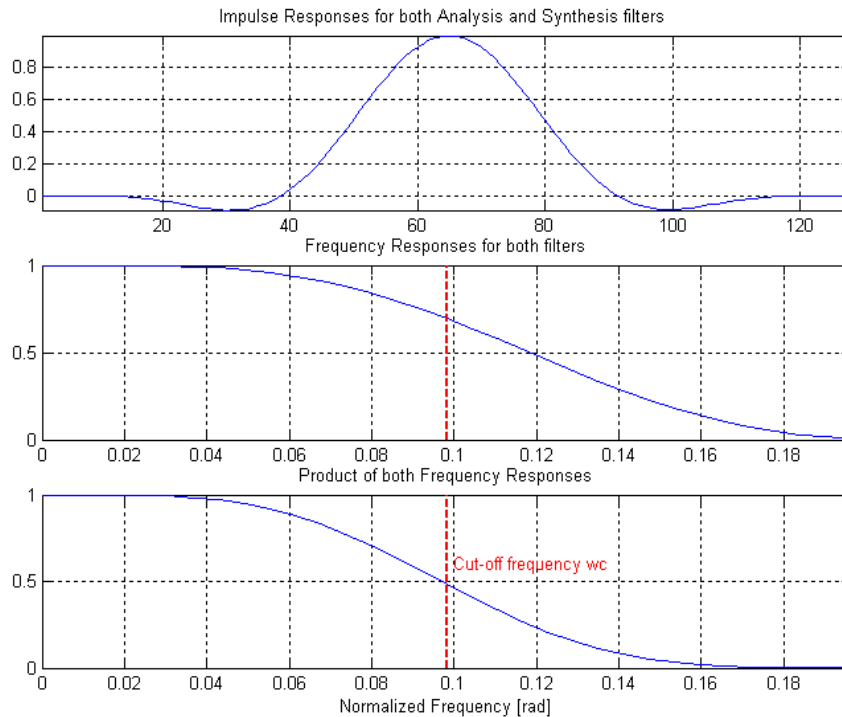
0.5, at frequency  $\omega_c$ . During synthesis, when adding consecutive bands together (Step 4 in Figure 7), the result now sums at 0.5 instead of 1. This means that huge ripples will be produced, reaching  $-6$  dB at the cut-off frequency. As a consequence, our filters are not anymore appropriate. When  $DF = 1$ , filters with a frequency response value close to the square root of 2 (0.707) at frequency  $\omega_c$  should be used, as the one illustrated in Figure 20. ON Semiconductor does not propose such filters as default when  $DF = 1$ , but they can be obtained contacting the support team. An example of design for such filters will be discussed in the next section, compromising the amount of ripples in the passband with the filter sharpness (Configuration 8).



**Figure 18. Impulse Responses (First Plot) and Frequency Responses (Second Plot) of Analysis and Synthesis Filters, in the Case  $L_a = 128$ ,  $L_s = L_a/DF = 64$ ,  $N = 32$**



**Figure 19. Impulse Responses (First Plot) and Frequency Responses (Second Plot) of Both Analysis and Synthesis Filters, in the Case  $L_a = 128$ ,  $L_s = L_a$ ,  $N = 32$**



**Figure 20. Impulse Responses (First Plot) and Frequency Responses (Second Plot) of both Analysis and Synthesis Filters, in the Case  $L_a = 128$ ,  $L_s = L_a$ ,  $N = 32$**

The third plot shows the product of both frequency responses. In this plot, another filter is used, having frequency response equal to 0.707 at the cut-off frequency

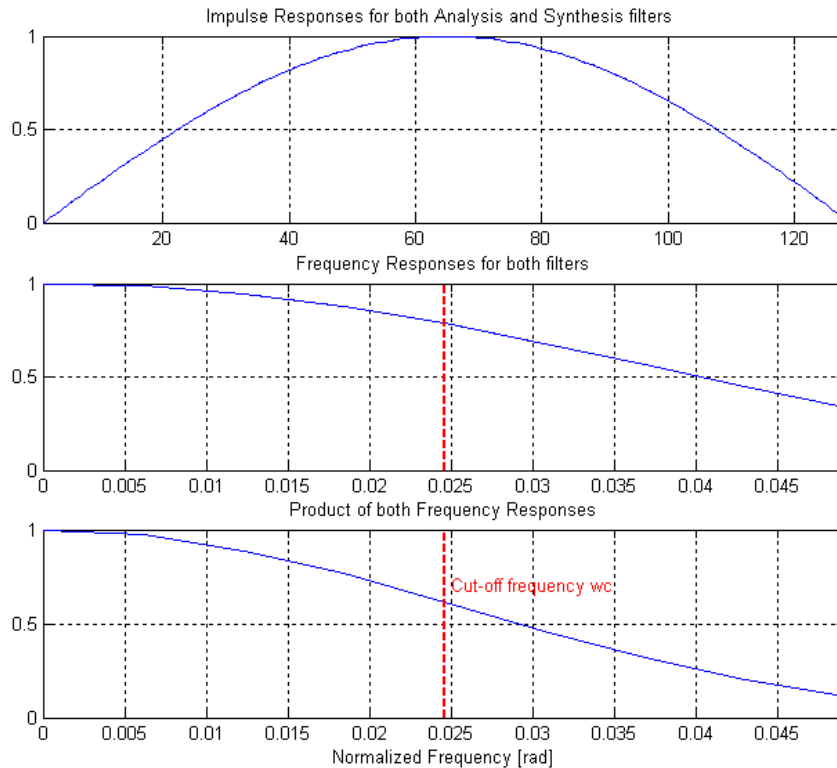
**Overlap–Add Configurations**

When the number of channels  $N$  is selected to be equal to the filter lengths  $La$  (the synthesis filter has generally the same length  $Ls = La$ ), the processing performed by the filterbank is similar to a short-term discrete fourier transform, (STDFT). In such cases, and if choosing the appropriate filters, it happens that both filterbank and STDFT interpretations are identical, as detailed in the next chapter. As a reminder, in those STDFT structures, which are often used in audio coding schemes, the overlapping input blocks are first windowed, typically with a Hann window. Then an FFT is performed. At synthesis, the inverse FFT is calculated, and the output block is finally reconstructed by multiplication with a rectangular window, and “overlap–add” operation. Alternatively, the input window could be the square root of a Hann window, a half–sinusoid. If this same window is also used for reconstruction, the square–root windows multiply together producing an overall Hann window. In comparison to using a Hann window only at analysis followed by a rectangular window at synthesis, this has certain advantages for minimizing distortion incurred during the gain–application and synthesis steps.

When programming the WOLA co–processor on the hardware, in all overlap–add configurations, the default analysis and synthesis filters impulse responses  $h(n)$  and  $f(n)$

are square–rooted Hann windows. If a synthesis is immediately performed after the analysis, without any processing in between, then the passband ripples are totally suppressed. However, as a counterpart, it can be observed on the last plot of Figure 21, that the frequency response of one band is enlarged compared to previous figures. Effectively, the frequency response of the prototype low–pass filter has still not reached 0 at frequency  $2\omega_c$ , which corresponds to the center of the adjacent band (at the very right end of the plot). As a consequence, in a general way, aliasing properties can be expected to be worse in overlap–add configurations. It should also be noted that with the sinusoid lobe impulse responses, the product of the analysis and synthesis frequency responses overpass 0.5 at the cut–off frequency. As a consequence, when summing adjacent bands, the amount exceeds 1. Despite of that, no ripples are produced. In effect, at the center of the band, the same amount is reached after summation, because of the non–zero components coming from both adjacent channels. The output signal level is scaled up.

Finally, let’s mention that applying the usual window filter design is not really appropriate in the overlap–add configurations. When  $N = La$  or  $Ls$ , it could be verified that the sinc expression  $h_o(n)$  is truncated in a much too rough way. This situation expresses the fact that the filter length is too short in respect to the number of bands.



**Figure 21. Impulse Responses (First Plot) and Frequency Responses (Second Plot) of Both Analysis and Synthesis Filters, in an Overlap–add Configuration with  $La = Ls = N = 128$**

The third plot shows the product of both frequency responses. In this plot, both impulse responses are sinusoid lobes.

## FILTERBANK DESIGN: PLAYING WITH THE FILTERBANK PARAMETERS (USING THE WOLA TOOLBOX)

After going through theoretical aspects in the previous chapter, this section deals with more practical considerations, in the goal of designing specific filterbanks. The WOLA toolbox for MATLAB is used as a tool to play with the different parameters, observing their effects on the filterbank characteristics. Interpretations of the observed behaviors are given, in respect to the principles described above. The goal of this chapter is for the reader to start “feeling” the filterbank, and then be able to select an appropriate configuration for his application. All listed examples were selected for didactical purpose only. They are not to be considered as models or “best solutions”, because the ensemble of appropriate configurations is generally wide, and judging the quality of a configuration always depends on the particular constraints of the application. Most of the following examples are built with the hearing-aid industry constraints in mind. The sampling frequency is always assumed to be 16 kHz, as it is generally in this context. Particular topics are discussed in more details, like overlap-add configurations or special filter design for  $DF = 1$ . At the end of the chapter, solutions to perform isolated FFTs are described.

### Typical Hearing-aid Configurations (16 bands)

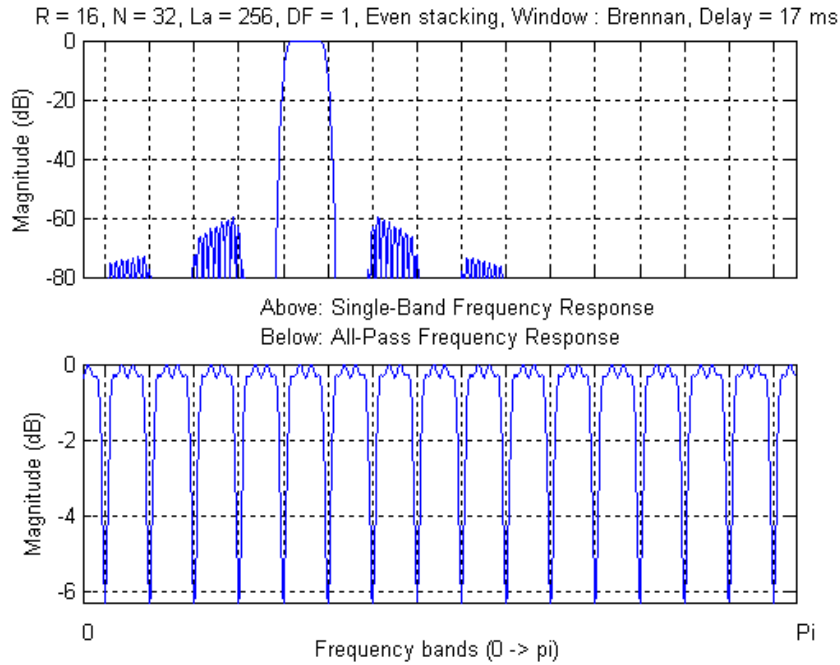
#### Configuration 1

Let's start this section with a first example of filterbank design, where the number of bands is specified to 16, which sets  $N = 32$  channels. Such a choice is often encountered in hearing-aid applications. Then, in order to choose the block size  $R$ , one may think about the following compromise: a longer block  $R$  allows for more algorithm operations in the block because of more processing cycles available at fixed DSP clock rate. Conversely, some algorithms need a fast parameter update rate, requiring a short block. For now, let us select  $R = 16$ . In the next step, the filter lengths will be addressed. Remembering that the longer the filter, the better the audio quality, one may select the longest available filters in the WOLA framework, that is  $L_a = L_s = 256$ . The synthesis filter is the same length as the analysis filter and,

consequently,  $DF = 1$ . It is important to remember that, as a default on the hardware, both filters would be built using the Brennan window, even though  $DF = 1$ . Selecting the even stacking mode, the WOLA filterbank (analysis + gain application + synthesis processes) would provide the frequency responses shown on Figure 22, when a unit impulse is presented at input. According to the even stacking mode, one can notice the DC and Nyquist bands at both extremities of the plots, with half bandwidths, compared to all other bands. For ease of comparison, all the frequency response plots are shown with vertical grids corresponding to the band edges while the frequency axis covers the whole range from 0 to  $fs/2$  or  $\pi$  (noted Pi).

The figure actually shows two frequency responses corresponding to different settings in the gains applied between the analysis and the synthesis. In the first plot, all gains are set to 0, except for band number 6. This choice clearly shows the behavior of the filter at the transition between subsequent bands. Furthermore, it depicts the amount, shape and location of the imaging distortion produced over the whole output spectrum by the processing of this particular band. These images are actually distortions, which would be superimposed to other channels, if all gains were set to 1. Every channel produces similar imaged components, (shifted in frequency), all contributions are summed when all bands are preserved at the output.

The second plot in the figure shows the frequency response when all gains are equal to 1. In other words, it shows the ripples produced in the passband, when no frequency shaping is applied by the filterbank. This representation is very useful for evaluating the quality of the filter shape in the all-pass sense. In the particular configuration, very large ripples are present in the passband. The range of those ripples reaches 6 dB. As explained in the previous chapter, this is because the default filters have been used while  $DF = 1$  (and  $N$  is different from the filter lengths). According to Figure 18, such a combination causes the signal to be divided by two at all band transitions, which corresponds to a 6 dB energy loss. A better filter would have to be designed, if this configuration was to be kept. An example of such a filter design will be shown later.



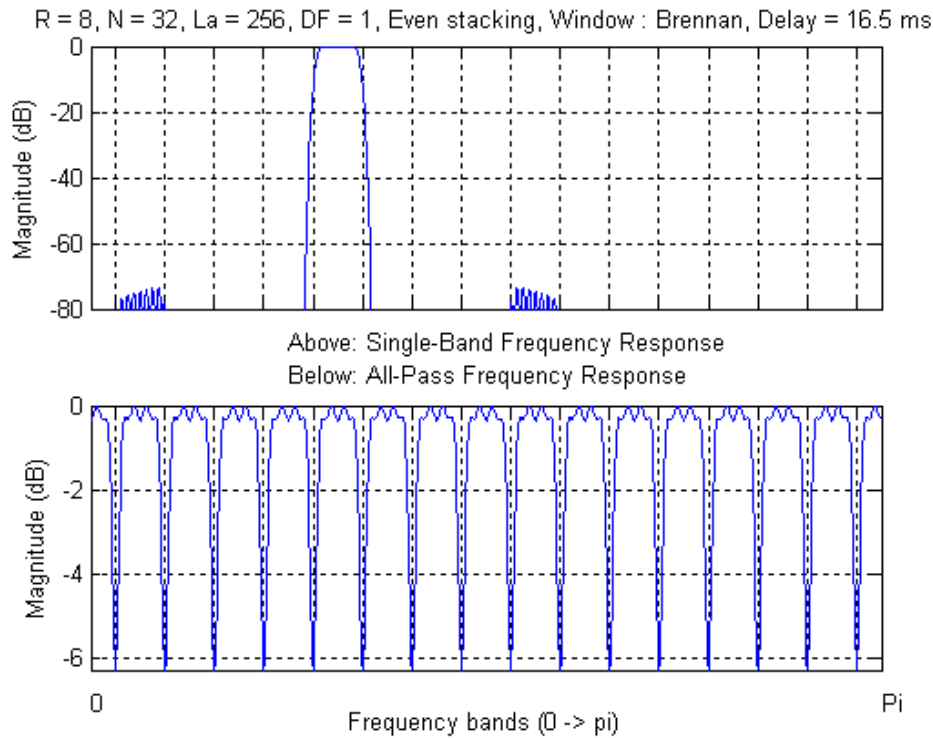
**Figure 22. Frequency Responses (Single Band-Pass and All-Pass) for the Specified WOLA Configuration (Configuration 1)**

This configuration shows a bad ripples behavior (6 dB ripples) because using the default filter when DF = 1. The aliasing level is lower than 60 dB, which is good however, the delay is too large.

Apart from this filter design issue, looking at first plot, the amount of imaging is observed to stay below -60 dB, which is very good. Furthermore, the transition of the filter is very sharp. Those are very good properties. However, a major problem could appear (especially in hearing aid applications), when considering the group delay, which is 17 ms (assuming 16 kHz sampling frequency). This is much too high in the hearing-aid industry.

**Configuration 2**

In hearing aids, the delay is generally to be kept lower than 10 ms. Consequently, the delay in our configuration should be largely decreased. As the delay (in terms of number of samples) is calculated according to the expression  $La/2 + Ls/2 + R$ , decreasing  $La$ ,  $Ls$  or  $R$  should be considered. In order not to compromise quality by reducing filter lengths, let's see at first what happens when setting  $R$  to 8 instead of 16. Figure 23 shows the behavior with those new settings.



**Figure 23. Frequency Responses (Single Band-Pass and All-Pass) for the Specified WOLA Configuration (Configuration 2)**

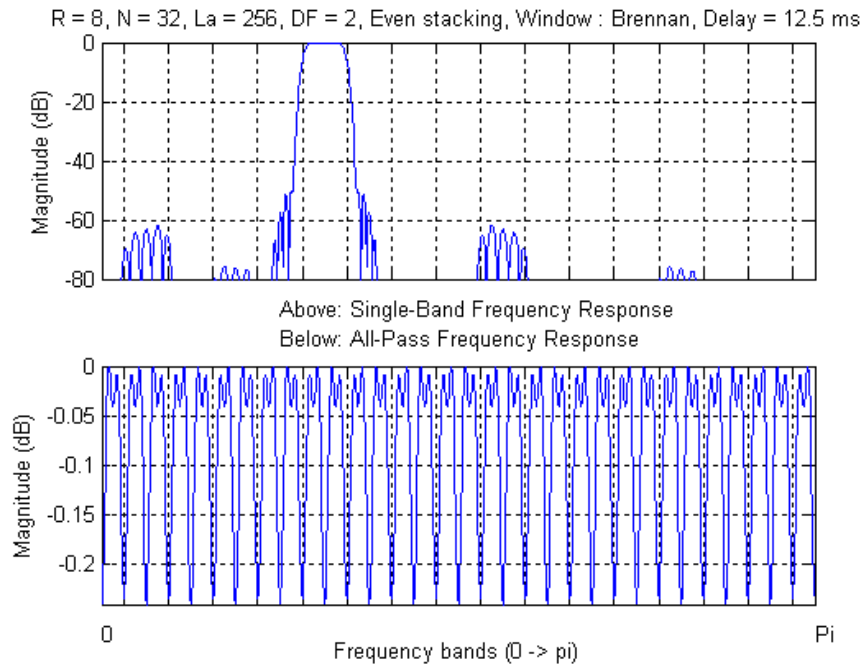
This configuration still shows a bad ripples behavior (6 dB ripples) because using the default filter when  $DF = 1$ . The aliasing is reduced compared to the previous figure, because the oversampling factor is higher. Delay is still very large.

The delay is not much shorter. Actually, in the particular configuration, reducing the block size  $R$  had only a slight effect because  $R$  is very small compared to the filter lengths. Looking at the plots, it can be observed that the ripples are still high, which is normal since  $DF$  is still one, and the filter was not changed. In first plot, however, an interesting event happened: the largest aliasing components, which were also the closest to the band of interest disappeared. The level of imaging is now lower than  $-70$  dB. What is the reason for this? Let's remember that the block size  $R$  in the WOLA scheme also represents the decimation in the filterbank. Considering again Figure 7 or Figure 11, one observes that when the decimation  $R$  is reduced, the images produced by the interpolation process in the synthesis are positioned farther away, one from the other. The number of those images is reduced and the distance between them,  $2\pi/R$  is now equal to  $\pi/4$  instead of  $\pi/8$  in Figure 22. This is a good point, which can also be expressed in terms of the oversampling  $OS = N/R$ , equal to four instead of two previously. Actually, the oversampling factor,  $OS$ , expresses the distance between the interpolation images in number of bands (number of times  $2\pi/N$ ). As a practical rule of thumb, it often turns out that selecting an oversampling factor equal to four is a good choice.

Talking a while about processing cycles, one must be aware that a configuration like the present one, using long filters, and having a relatively short block size  $R$ , represents a lot of calculation in the coprocessor. As long as the number of cycles do not overpass the total number of processing cycles available during one block (which depends on the selected DSP clock frequency), this may not be a concern, provided that the processors (WOLA and RCore) are used in parallel according to the "one-frame-delay" scheme (described in ON Semiconductor's literature). However, one should be aware that the DSP consumption will be higher.

### Configuration 3

The second configuration discussed above still suffers from a too large delay, and this must be improved, reducing the filter lengths. Thinking of the above observations about interpolation images, one realizes that, as the distance between those images is now two times larger, a filter with smoother transition slope (that is a shorter filter) could be used at synthesis. This should not increase the aliasing in a critical way, and the delay would be reduced. In this optic, let's see what happens changing  $DF$  to 2:



**Figure 24. Frequency Responses (Single Band-Pass and All-Pass) for the Specified WOLA Configuration (Configuration 3)**

This configuration has better ripple behavior, lower delay and still good aliasing properties.

Looking first at the second plot, it is observed that the default filter built with the help of the Brennan window is now well adapted: the dynamic range of the ripples has decreased to values lower than 0.2 decibels. Then, in the first plot, one notices that imaging is more present than before, although still contained to an acceptable level. Actually, part of the aliasing generated at analysis has now an increased level, even though the analysis filter has not changed. This is partially caused by the reduced sharpness of the synthesis filter, which previously better helped removing those components. It must be understood there, that the synthesis filter, while mainly used to remove the images produced at synthesis, also performs a second filtering pass on the remaining aliasing components produced at analysis. The influence of the shorter synthesis filter can be observed both at the transition of the band of interest (slopes are reduced) and by the presence of higher/new aliasing components, for example on the left of the band, at the position previously occupied by the first interpolation image.

For illustration purposes, the vertical axis range of the figures are usually set in a way to show the frequency responses level between 0 dB and -80 dB, displaying only

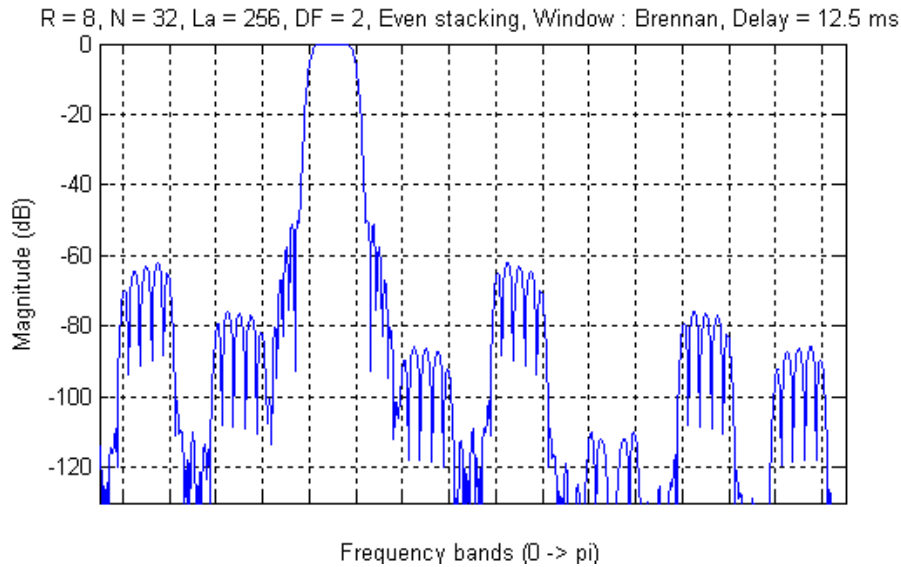
significant noise components. However, looking at the same plot (the upper plot of Figure 24), while extending the vertical scale towards lower levels (Figure 25), it becomes obvious that aliasing and imaging are actually present everywhere.

The aliasing from analysis is particularly present near the band of interest, while the synthesis imaging level is evidently higher at the position where the images are located. During synthesis, the remaining amount of aliasing produced at analysis is actually imaged to other locations in frequency, and the final aliasing distribution is quite complex. Fortunately, this imaging of the aliasing distortion is an extremely small effect since both analysis and synthesis filters act to reject it and it may be neglected.

Changing the mode to odd stacking (Figure 26), one can observe that aliasing and imaging becomes differently distributed. This is because the respective imaging components produced at reconstruction of the particular channel and of its complex-conjugate have now different relative positions. This results in slightly different distortions properties, which are sometimes preferred.

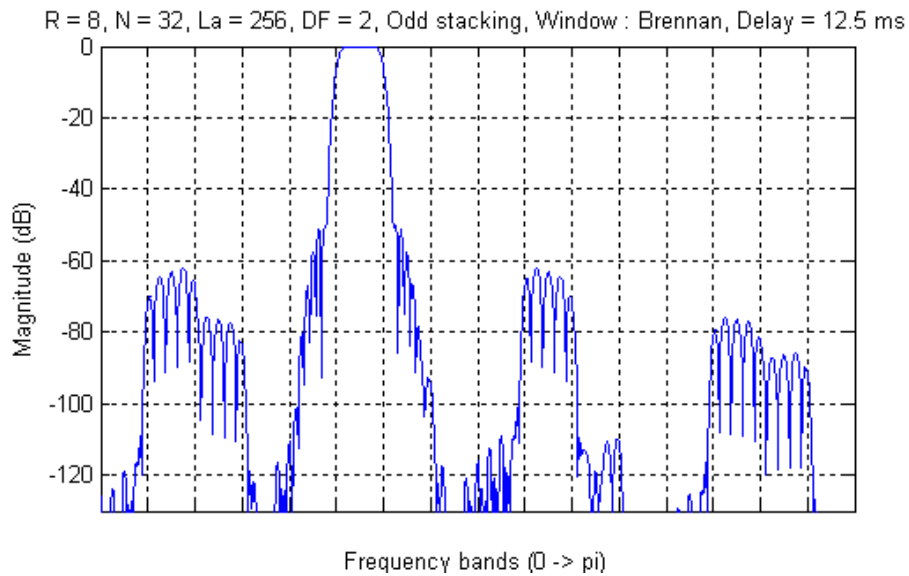


## AND8382/D



**Figure 25. Frequency Response (Single Band-Pass) for the Specified WOLA Configuration**

Aliasing is everywhere, but images produced at synthesis are dominant.



**Figure 26. Frequency Response (single band-pass) for the Specified WOLA Configuration, Which is the Same as Above, Except that Odd Stacking is Selected**

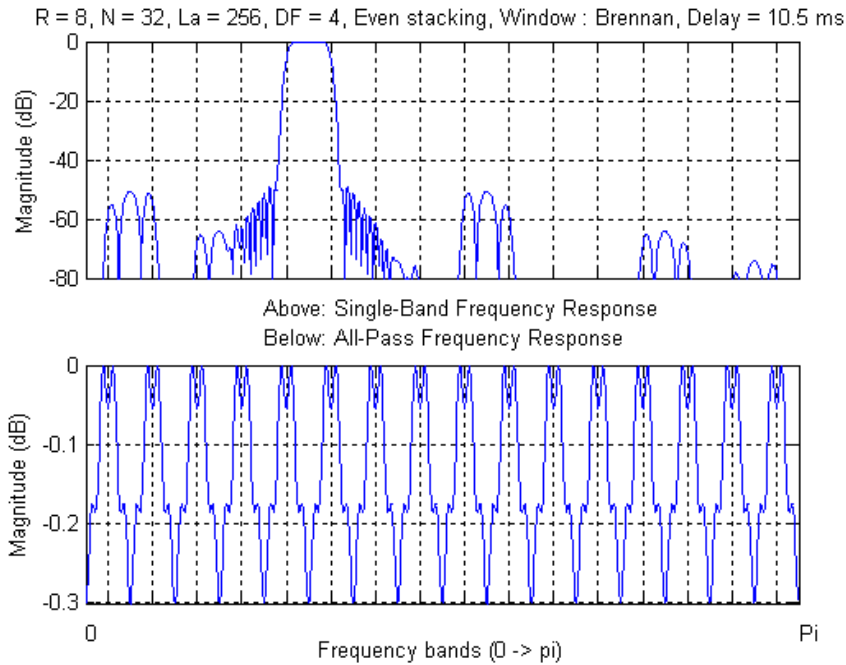
Aliasing and imaging distribution is different.

As a conclusion, this third configuration allowed to importantly decrease the delay, while preserving aliasing components below -60 dB, which is still very good. Unfortunately, our 10 ms goal for the delay is still not reached.

**Configuration 4**

In this fourth example, the synthesis filter length is furthermore reduced to  $L_s = 256/4 = 64$ , setting  $DF = 4$ . Instead of 12.5 ms, the delay gets now to 10.5 ms, which is

unfortunately not low enough. Figure 27 illustrates the associated frequency responses. No particular modifications appear in ripples level, while aliasing is still acceptable, even though it gets closer to  $-50$  dB.



**Figure 27. Frequency Responses (Single Band-Pass and All-Pass) for the Specified WOLA Configuration (Configuration 4)**

This configuration still shows acceptable aliasing properties, but delay still overpass 10 ms.

**Configuration 5**

Furthermore increasing  $DF$  to 8 would not be a good way to lower the delay, as only 1 ms would be gained. In addition, it would make the cut-off frequency of the prototype low-pass synthesis filter reach eight times the cut-off frequency of the analysis filter, that is  $8\omega_c = 8/N = 8\pi/32 = \pi/4$ . This would cause the synthesis filter frequency response equal to 0.5, at  $\pi/4$  radians away from the center of the band of interest. As the first image to be removed by the synthesis filter is precisely centered at that position ( $OS$  being 4, as discussed in Configuration 2), an unacceptable imaging situation would occur. In general, for that reason, selecting  $DF$  higher than  $OS$  is not possible.

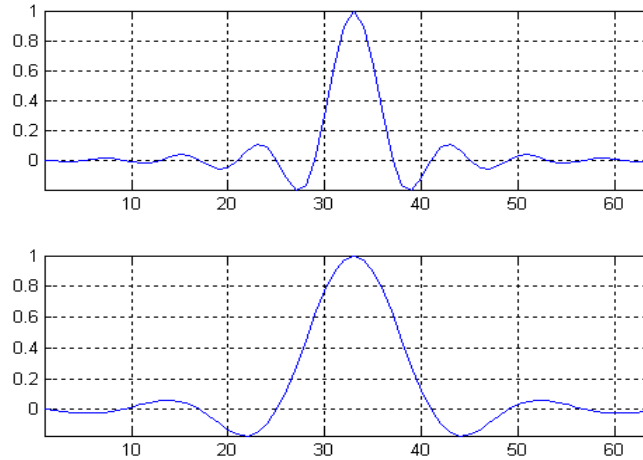
Therefore, the only option left for reducing the delay is to scale down the analysis filter length by 2, selecting  $L_a = 128$ . In order to keep the same synthesis filter length as in the previous configuration ( $L_s = 64$ ), let's set  $DF = 2$  instead of 4. Figure 27 shows the associated frequency responses. As a result, the delay is now 6.5 ms, while the amount of aliasing is reduced compared to Configuration 4. It is effectively back around  $-60$  dB. This configuration appears very interesting.

At that point, let's emphasize what happens when one wants to decrease the analysis filter length (typically for the

purposes of reducing delay) for example by a factor of two, while leaving the synthesis filter length unchanged. Since the synthesis filter is obtained by decimation of the analysis filter, one is tempted to simply divide  $L_a$  by 2, and divide  $DF$  by 2 as well. This is just what was done in the example above. One should beware: while the synthesis filter is effectively the same length as before, its shape and properties are no longer the same. Effectively, thinking about the way the analysis filter is built from the sinc function (see section about prototype filter design), it happens that a synthesis filter obtained by decimation (factor 4) of a 256 coefficients analysis filter is not the same as a synthesis filter obtained by decimation (factor 2) of a 128 coefficients analysis filter.

In fact, for the same number of channels  $N$ , the impulse response of the 256 coefficients analysis filter has twice as many zero-crossings as the impulse response of the 128 coefficients analysis filter. Then, because of decimation, this becomes also true for the synthesis filter produced from the first analysis filter (256 coefficients), compared to the synthesis filter produced from the second analysis filter (128 coefficients). Both have same lengths (64 coefficients in this example), but the second one has less zero-crossings. In fact, the main lobe of its impulse response is wider, which causes its frequency response to be narrower.

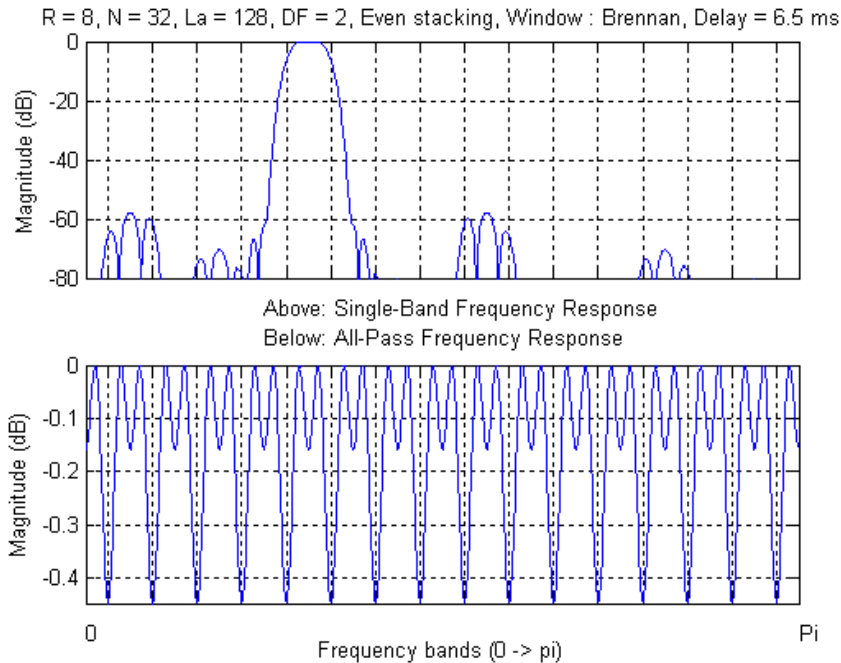
## AND8382/D



**Figure 28. Generation of the Synthesis Filter Prototype as a Decimated Version of the Analysis Prototype Filter**

The above plots show the situation. The first synthesis filter impulse response has length 64 and was generated from the 256 coefficients analysis filter. The second one has

same length, but was generated from the 128 coefficients analysis filter. In both cases,  $N$  was 32. In the second case, the bandwidth of the filter is reduced by 2.



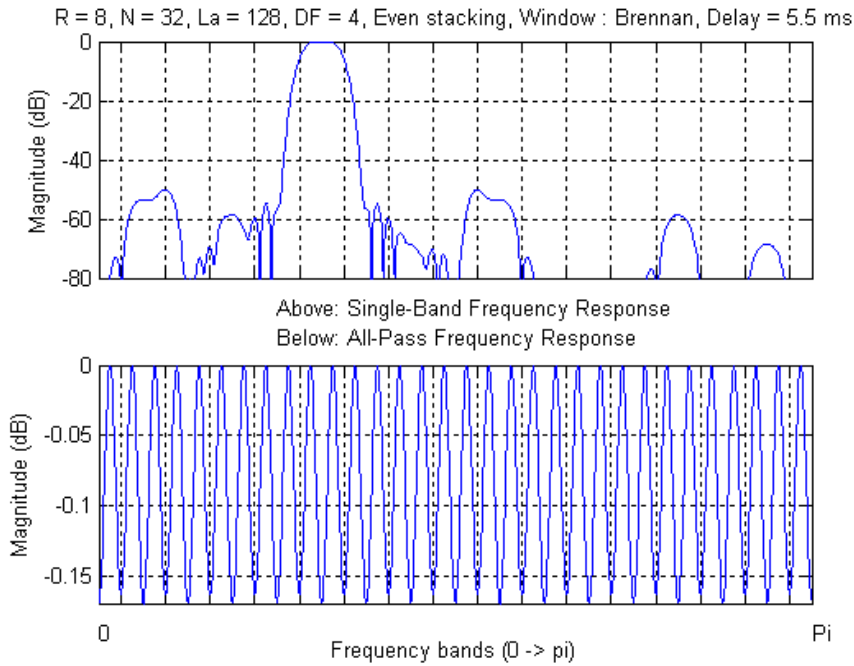
**Figure 29. Frequency Responses (Single Band-Pass and All-Pass) for the Secified WOLA Configuration (Configuration 5)**

This configuration finally reaches the aimed delay. The level of aliasing seems good, however, the transition area between bands is now enlarged (reduced slopes), which may be a problem when very different gains are applied in one band and its adjacent one.

**Configuration 6**

The delay in the previous configuration was good. However, there may exist situations in which it should even be lower. As observed on Figure 30, a 5.5 ms could be obtained, while reducing furthermore the length of the

synthesis filter to 32, setting *DF* to 4 instead of 2 (which is OK, as *DF* is still not higher than *OS*). Such an operation does not affect the noise level too dramatically, as it stays lower than -50 decibels.



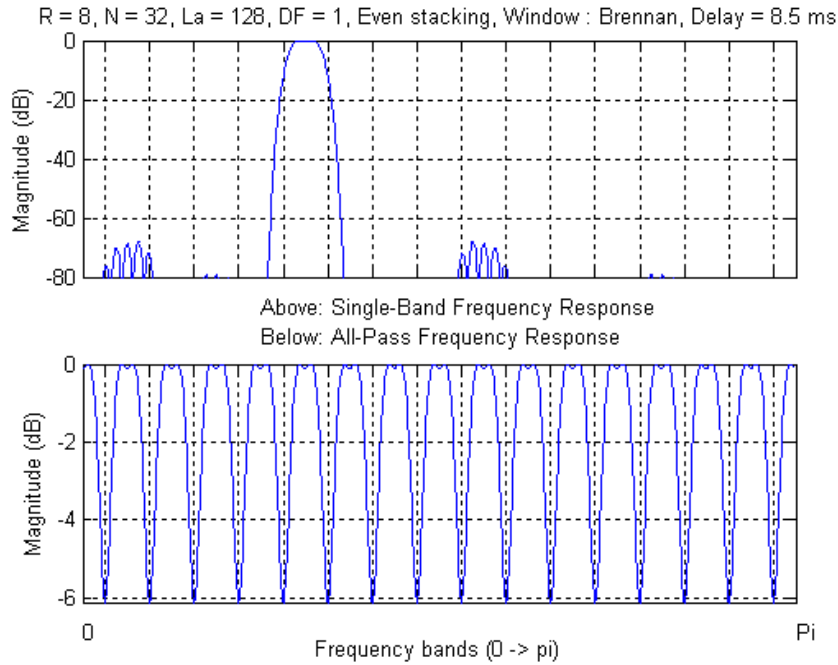
**Figure 30. Frequency Responses (Single Band-Pass and All-Pass) for the Specified WOLA Configuration (Configuration 6) Which Furthermore Improves the delay to 5.5 ms**

The level of aliasing is still acceptable.

While this configuration is having great success in hearing aids applications, one should notice that the slope of the global frequency response in the transition areas between bands is much smoother, when compared to Configuration 4 and previous. As a consequence, when applying frequency shaping with sharp characteristics (big differences between one gain value and the one associated to the adjacent band), much more aliasing and imaging may be produced. Sometimes, a smoothing of the gain set may be performed in the frequency domain, prior to gain application. This remark holds both for Configurations 5 and 6.

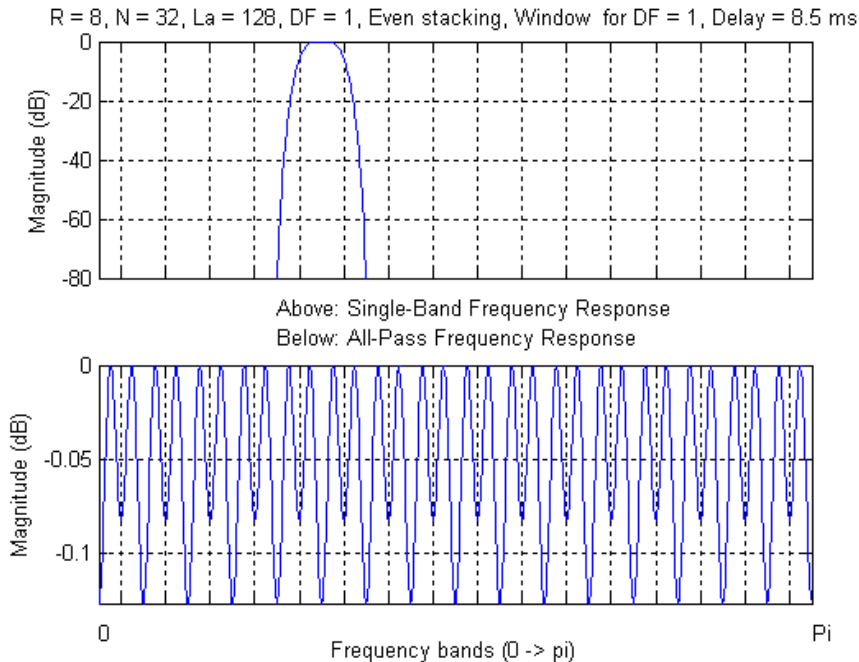
**Configuration 7**

Whenever the above remark represents a problem, *DF* could be set again to 1, featuring two filters of length 128. In such a case (Figure 31), the aliasing behavior improves incredibly, while the delay stays in the original amount specified: 8.5 ms. The sharpness of the frequency response is again good. However, as we used the default window, ripples reaching 6 dB are again observed.



**Figure 31. Frequency Responses (single band-pass and all-pass) for the Specified WOLA Configuration (Configuration 7)**

This configuration features very good aliasing properties (lower than -60 dB). Unfortunately, passband ripples are huge again because the default window was used.



**Figure 32. Frequency Responses (Single Band-Pass and All-Pass) for the Specified WOLA Configuration (Configuration 8)**

This configuration features good aliasing properties (lower than -80 dB but smooth gains required). Passband ripples problems are solved due to a specially-designed window

The time has now come to tackle the ripples issue, observed when  $DF = 1$ . As exposed in the previous chapter and illustrated in Figure 19, the reason for those huge ripples originates in the fact that the product of both analysis and synthesis impulse responses equals to 0.25 at the cut-off frequency  $\omega_c$  instead of 0.5 in other configurations. As a result when summing two adjacent bands during output signal reconstruction, both components sum at 0.5 (–6 dB energy loss) instead of summing at 1.

A solution consists of designing filters having frequency responses values reaching 0.707 at the cut-off frequency. In such a way, their product would equal 0.5, and the reconstruction would sum at 1. In order to do that, looking at the second plot in Figure 19, one could simply manage to push the decay of the responses a little bit on the right, in order for them to reach 0.707 at frequency  $\omega_c$ . This would correspond to a slight enlargement of the frequency responses, compromising the aliasing properties, but reasonably. In the time-domain, the impulse response of those filters (first plot in Figure 19) would have to be compressed, according to the time-frequency duality.

Considering that the impulse responses of the filters are generated by multiplying a window and a sinc function  $h_o(n)$ , both components could be affected to reduce the largeness of the impulse response main lobe. Either another window could be designed, or alternatively, keeping a fixed window, the sinc function could be slightly modified. In the present example, the second option is retained. Let's remember that the sinc expression is  $h_o(n) = \text{sinc}((n-La/2)/N)$ , which is equal to zero at every multiple of  $N$  samples (starting from the center of the main lobe at  $n=La/2$ ). Therefore, the largeness of the main lobe could be reduced by slightly increasing  $N$  in the expression for  $h_o(n)$ . As expected, this actually corresponds to a design for a larger cut-off frequency (as  $\omega_c = \pi/N$ ).

Choosing  $w_o(n)$  as the Hann window, such a method was applied in our present configuration (where  $N = 32$ ). The default window  $h_{\text{def}}(n)$  was replaced by  $h_{\text{new}}(n)$ , whose shapes and frequency responses are compared in Figure 33, as well as all their respective construction processes.

$$h_{\text{def}}(n) = w_o(n) h_o(n) = \text{Brennan}(n) \text{sinc}((n-La/2)/N)$$

$$h_{\text{new}}(n) = w_o(n) h_o(n) = \text{Hann}(n) \text{sinc}((n-La/2)/26.22)$$

Figure 32 shows the performance of such a configuration. The ripples are reduced a lot, while the aliasing behavior is further improved. No more aliasing is present above –80 dB. Such a configuration is a very satisfying solution, when delay should be lower than 10 ms, while not as low as around 5 ms. Very few aliasing will be produced, provided that smoothed enough frequency shaping schemes are used: the filter slopes, while better than in Configurations 5 or 6, are not so sharp (compare Figures 31 and 32).

**Configuration 9**

As a last example for 16 bands configurations, let's mention that when really low-delay are required, the filter length can be further decreased. Setting  $La = 64$  instead of 128 in Configuration 5, one can decrease the delay to 3.5 ms, while keeping an acceptable amount of aliasing. The price to pay is the very large band response, as shown on Figure 34. However, taking great care about the frequency gain shaping strategy (smoothness) ensures proper behavior. If this is not possible, then  $DF$  could be set to 1, getting sharper transitions, and 4.5 ms delay.

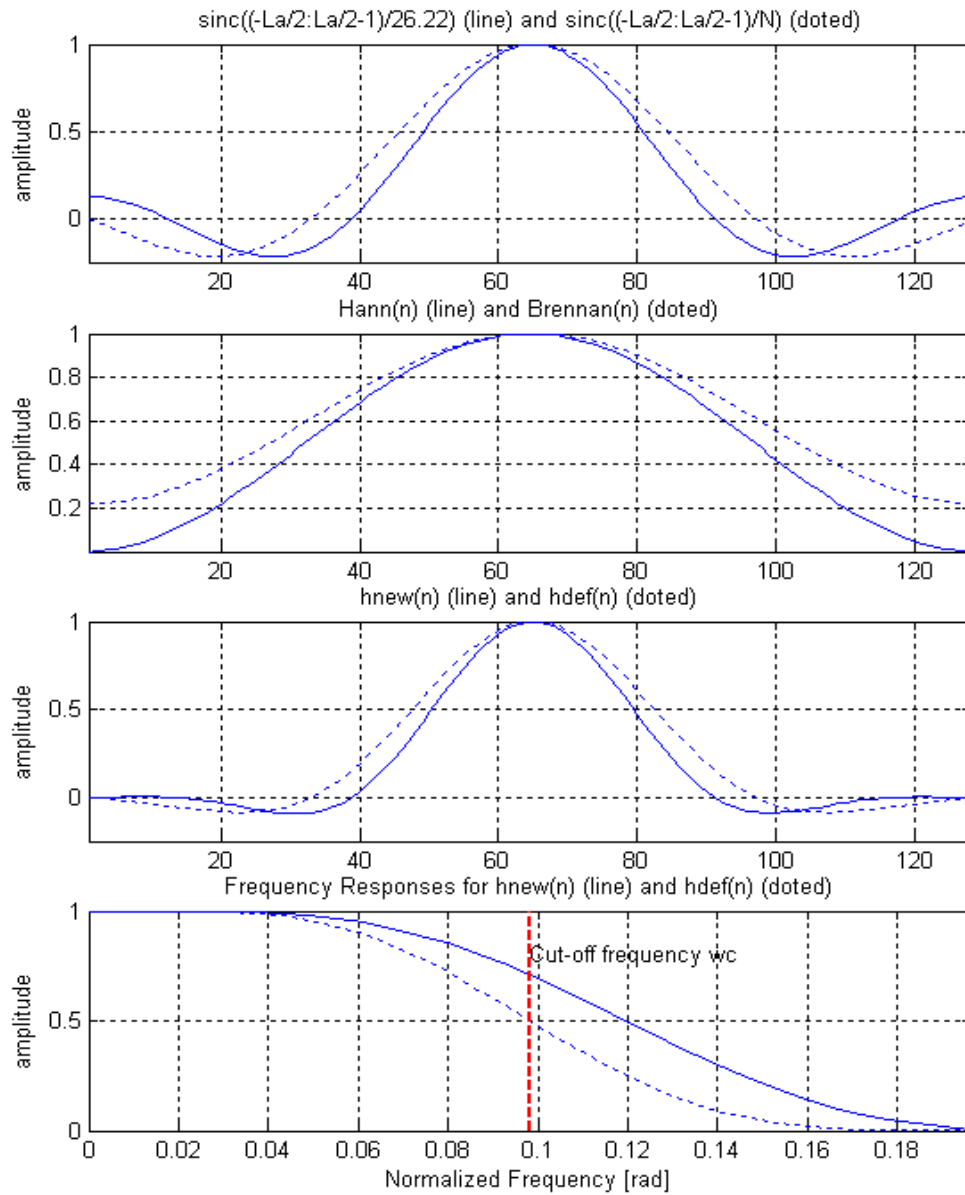
**Refined Frequency Resolutions**

This second set of configurations deals with improved frequency resolution. Examples for 32, 64 or 128 bands are discussed. Generally, getting to refined frequency resolution while keeping the same amount of aliasing (same audio quality) requires better filters, that is longer impulse responses and longer delays. Of course, if the sampling frequency can be increased, the delay can be kept reasonable, at the price of more processing power and consumption. It is a matter of compromise. The challenge is to find the best compromise depending on particular application constraints.

As this section is dedicated to a more extended range of applications, which do not always assume  $F_s = 16$  kHz, let's remember that the delay is function of the filter lengths, but above all of the sampling frequency  $F_s$ :

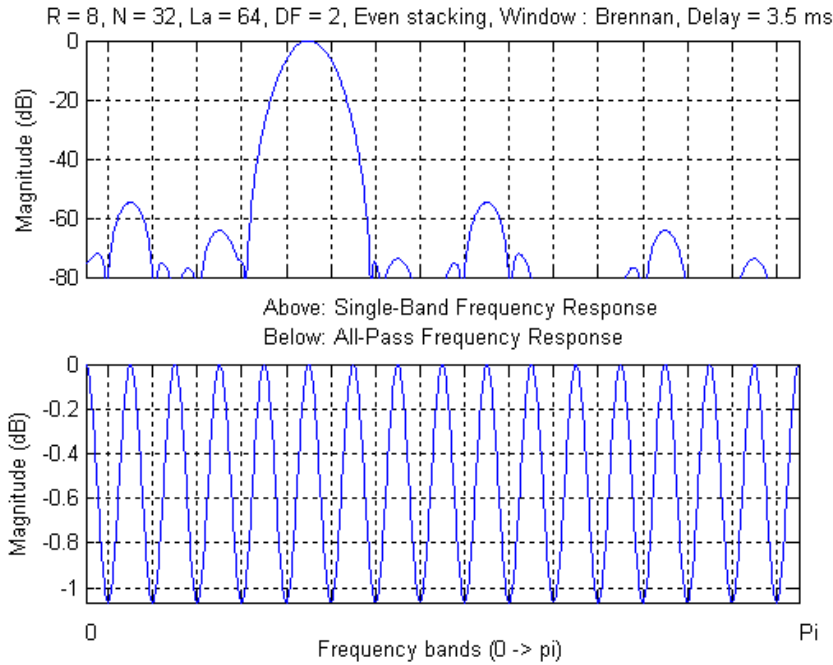
$$\text{Delay} = (La/2 + Ls/2 + R)/F_s$$

All mentioned delays will still be expressed for 16 kHz rate, but one should be aware that an 8 kHz sampling rate would double this value, as 32 kHz would make it half.



**Figure 33. Construction of the New Filter, in Order to Reduce the Ripples in Configurations with DF = 1**

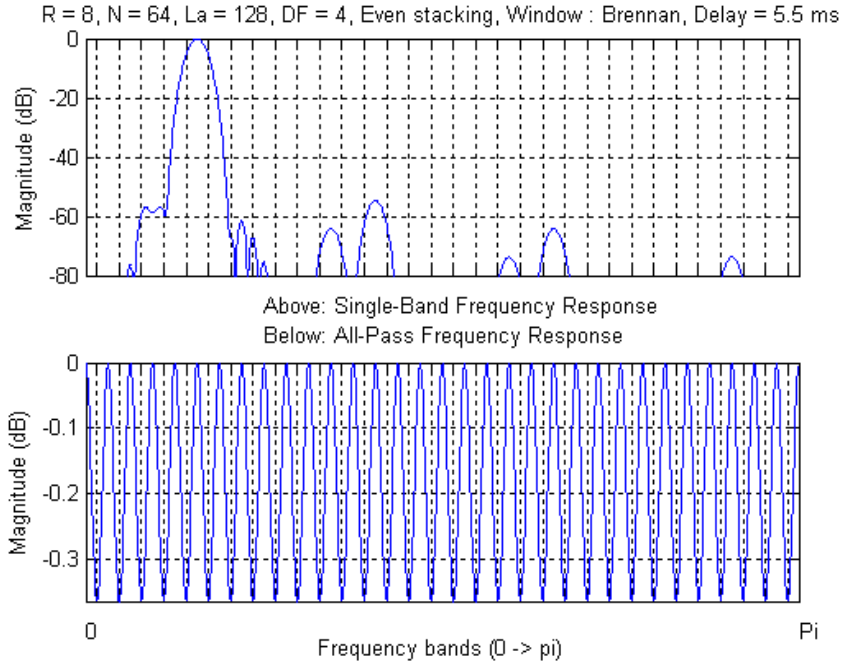
The dotted plot corresponds to the default filter. It can be observed that the frequency response of the new filter has value 0.707 at the cut-off frequency, instead of 0.5. As a counterpart, its sharpness is decreased. As shown in the third plot, this corresponds to an impulse response with compressed main lobe.



**Figure 34. Frequency Responses (Single Band-Pass and All-Pass) for the Specified WOLA Configuration (Configuration 9)**

This configuration is dedicated to very low delays, while keeping satisfying level of aliasing. The price to pay is the very large transition area between bands, requiring highly smoothed gain shapes.

**Configuration 10**



**Figure 35. Frequency Responses (Single Band-Pass and All-Pass) for the Specified WOLA Configuration (Configuration 10)**

The delay and level of aliasing are very satisfying. As a compromise, the transition area is large, preventing the use of very sharp gain curves.

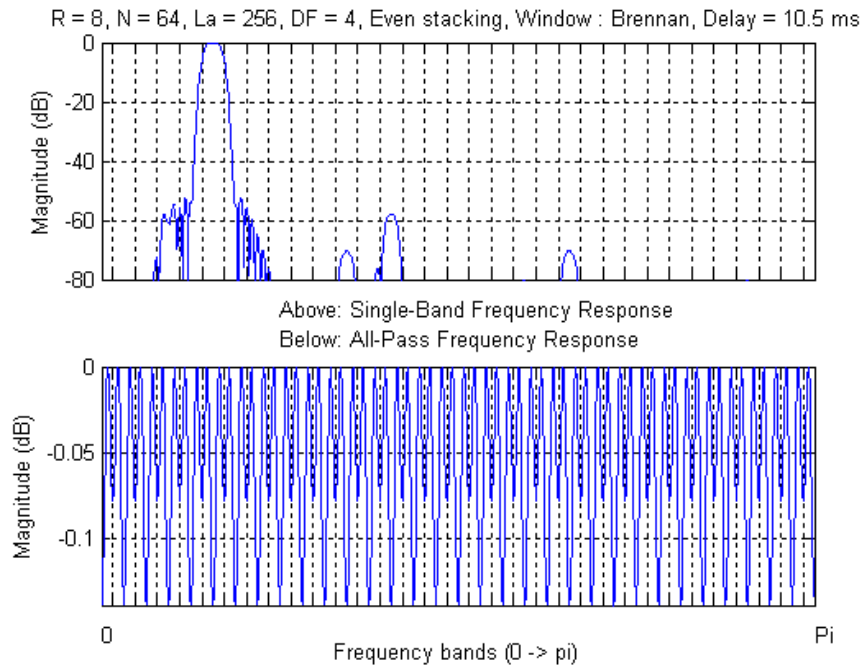


Starting from Configuration 6, which was satisfying enough when  $N = 32$ , let's modify  $N$  to 64, in order to get 32 bands. This is shown in Figure 35, which shows a very satisfying level of aliasing, with an impressively low delay (5.5 ms) for such a frequency resolution. However, the compromise appears when considering the lack of filter sharpness, which prevents gain shaping curves with significant sharpness. This is maybe not as annoying as in 16

bands situation, and could be a good solution depending on the algorithm. Let's see if improvements are possible, however.

#### Configurations 11A and 11B

In order to sharpen the filter transition, let's increase  $La$  to 256.



**Figure 36. Frequency Responses (Single Band-Pass and All-Pass) for the Specified WOLA Configuration (Configuration 11A)**

This configuration has very good aliasing properties, while featuring 10.5 ms delay, which may be acceptable (if not, 9.5 ms can be reached, changing  $DF$  to 8, see Figure 37). Such configurations, while very good at aliasing level, are very power consuming. This may be a problem depending on the DSP clock.

This configuration is excellent considering aliasing properties. Unfortunately, the delay has jumped to 10.5 ms. This may be acceptable, taking account of the improved frequency resolution which is now available compared to previous 16-bands settings. If not,  $DF$  could be set to 8, without large modifications of the aliasing properties because the oversampling  $OS$  is presently 8 (this partly explains the good aliasing behavior). With  $DF = 8$ , the delay would therefore go down to 9.5 ms (assuming 16 kHz), under the 10 ms barrier. This is shown on (Configuration 11B)

Actually, the major problem of such configurations may be related to their complexities, which are very high, looking at the ratio  $La/R = 32$ . This is huge. Depending on the selected DSP clock, those configurations may not be

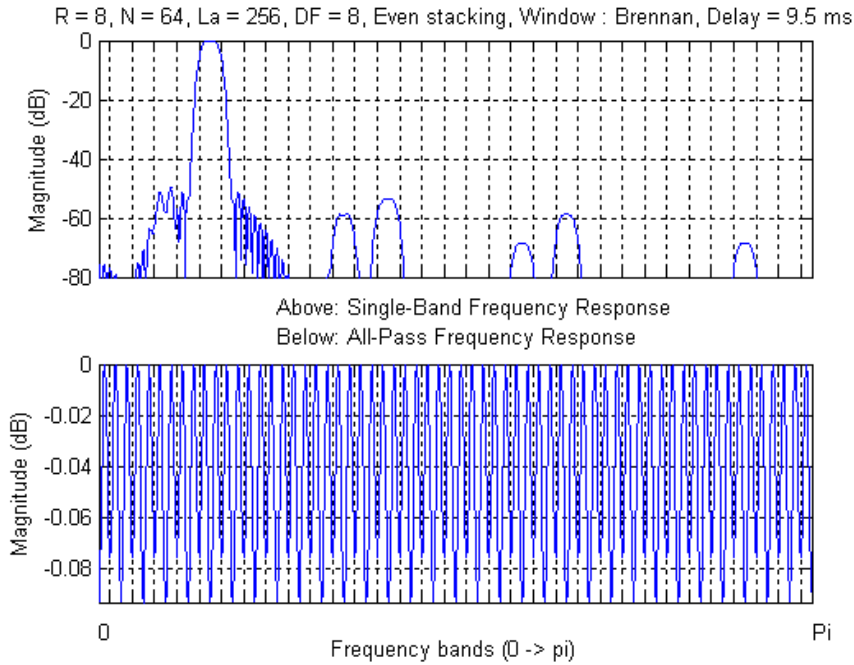
authorized. As a consequence, let's try and reduce this ratio, setting  $R = 16$ .

#### Configuration 12

When setting  $R = 16$ , the oversampling  $OS$  falls down to 4 again. As a consequence,  $DF$  can maximally be 4. As shown on Figure 38, the aliasing is still very good, particularly considering sharp transitions. The delay has increased again (to 11 ms at 16 kHz), which may again be acceptable for certain applications.

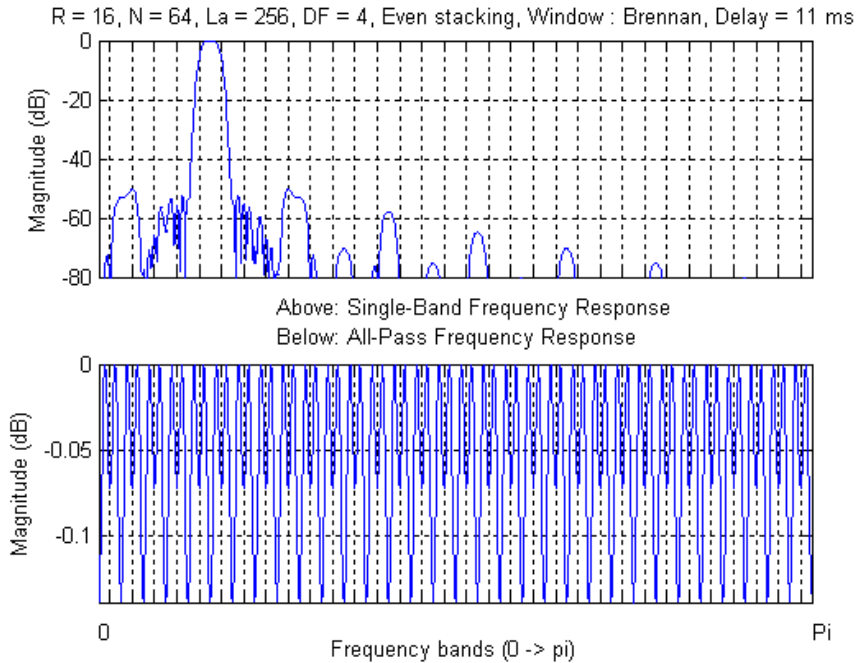
Obviously, among the configurations discussed here, the choice would depend on the application constraints: Configuration 10 (very low-delay, smoothed gains shaping curves), Configuration 11B (excellent aliasing properties and satisfying delay but high complexity), or Configuration 12 (low aliasing, fair complexity, but higher delay).

# AND8382/D



**Figure 37. Frequency Responses (Single Band-Pass and All-Pass) for the Specified WOLA Configuration (Configuration 11B)**

Slightly reduced delay and complexity compared to configuration 11A. Similar aliasing.

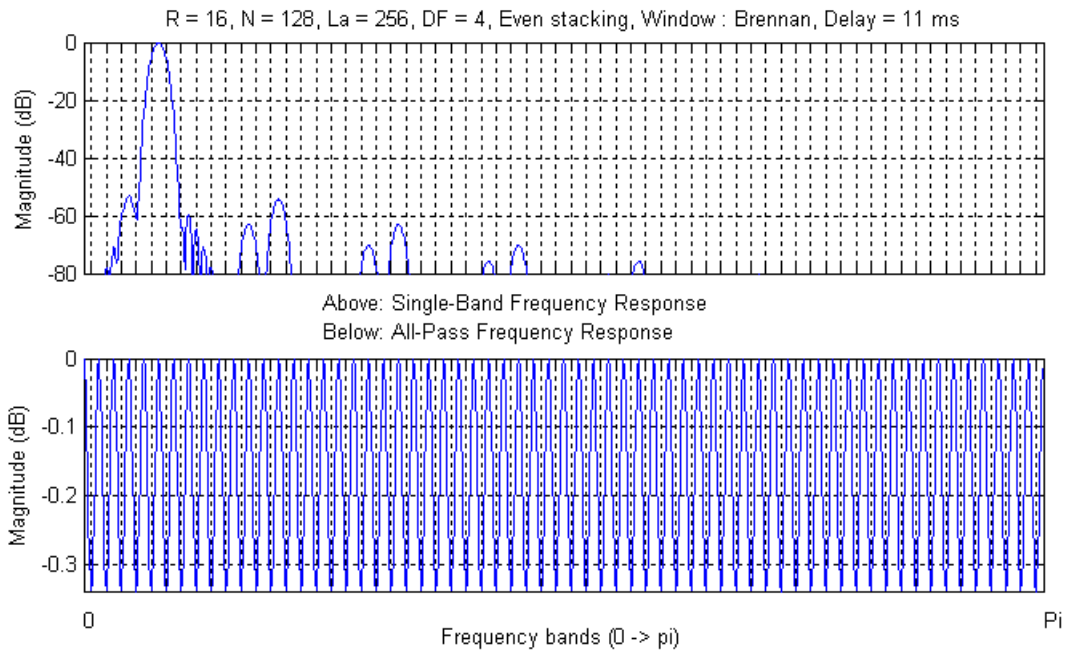


**Figure 38. Frequency Responses (Single Band-Pass and All-Pass) for the Specified WOLA Configuration (Configuration 12)**

Satisfying aliasing, higher delay. The complexity is reduced.

**Configuration 13**

Let's now further increase the frequency resolution to 64 bands, that is  $N = 128$ . Simply keeping other parameters unchanged compared to Configuration 12 provides the situation of Figure 39:



**Figure 39. Frequency Responses (Single Band-Pass and All-Pass) for the Specified WOLA Configuration (Configuration 13)**

This configuration provides a good compromise for 64 bands.

This configuration is far from being bad. It is clear, however, that the transition of the filters are getting smoother, in relation to the narrower bandwidths of each band. Consequently, as the filter lengths are limited to 256 in the hardware and cannot increase in proportion with the number of bands, smoother gain shaping curves have to be designed in the frequency domain. This is a general observation, to be taken into account in the WOLA, when  $N$  exceeds 64.

In the present configuration, if this delay (11 ms) is considered as excessive, one could try and reduce either  $R$ ,  $La$  or  $Ls$ . Let's try this as a next example.

**Configuration 14**

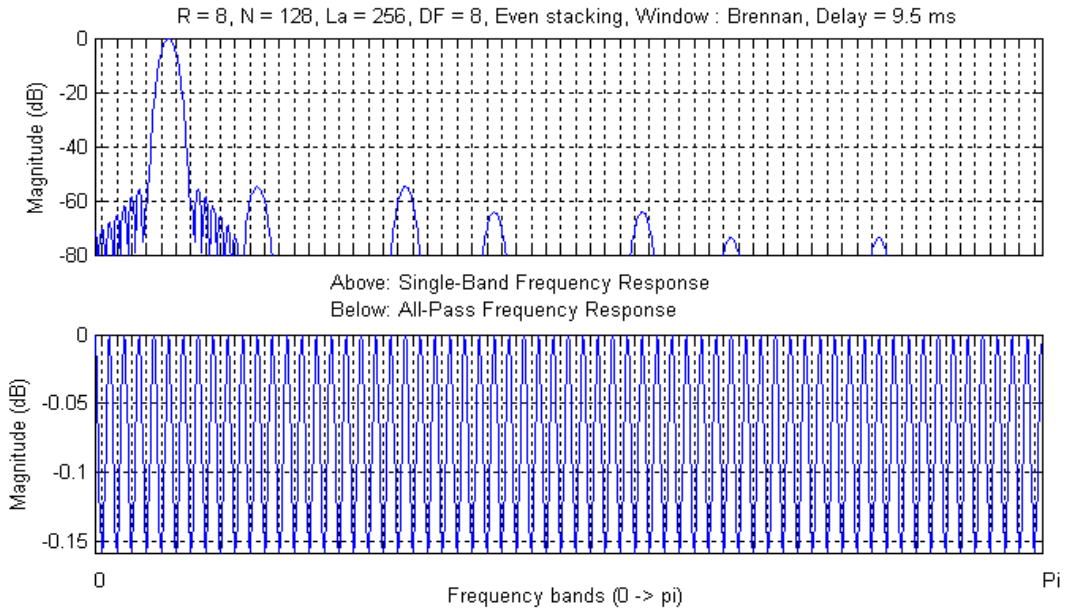
Assuming that the clock frequency allows for such an intensive configuration, one could set  $R$  to 8, keep  $La$  at 256 for best quality, and set  $DF$  to 8, which seems to be allowed as the oversampling factor  $OS$  is  $128/8 = 16$ . Such a configuration would certainly be satisfying, featuring advanced frequency resolution, satisfying audio quality and delay. This can be checked out in . Delay is 9.5 ms.

It is interesting to remark that in this configuration, the analysis filter is not selective relative to the narrow bands

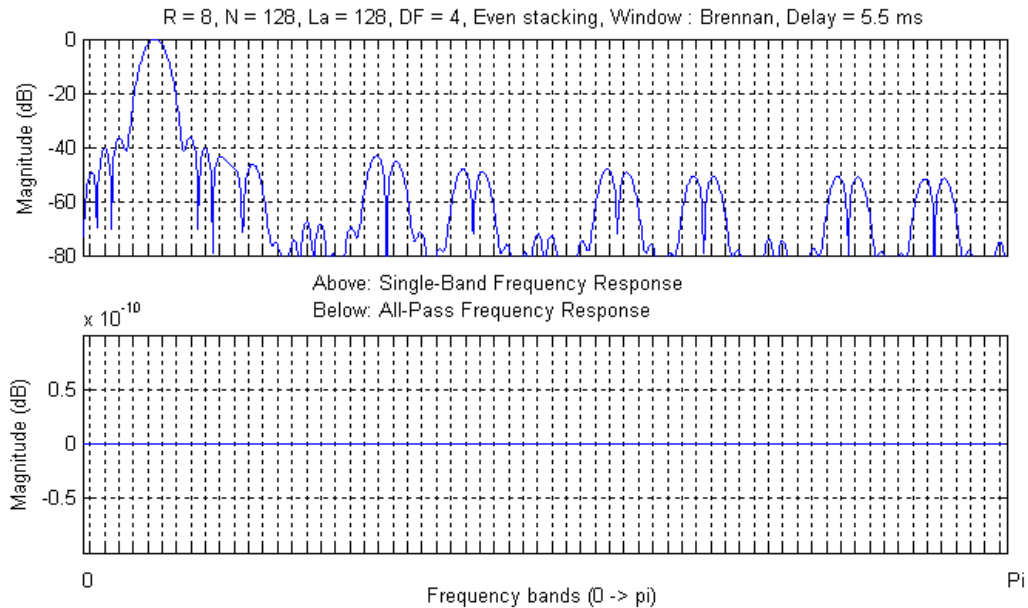
(the ratio between  $La$  and  $N$  is only 2, while it was 4 in all previous examples, except Configuration 9). As a result, the synthesis filter must not be too short, because it is also expected to participate to the elimination of aliasing remaining after analysis. Therefore, setting  $DF$  to 16 in the present situation would not provide good results. More generally,  $DF$  should be increased with caution when the number of bands is large.

**Configuration 15**

Following the usual way of thinking, in order to further reduce the delay, we could as well decrease  $La$  to 128. Of course, in order to keep the same synthesis filter length than before, we should assign  $DF$  to 4 instead of 8. Such a configuration would provide a 5.5 ms delay. Actually, for better aliasing properties, further reducing  $DF$  to 2 is interesting, while keeping the delay at 6.5 ms. The result is shown in Figure 41. The reduced analysis filter length makes the transition characteristics very poor in the frequency response. Great care should be taken with the gain settings. Furthermore, the imaging produced in one band is spread over to whole spectrum at high levels.



**Figure 40. Frequency Responses (single band-pass and all-pass) for the Specified WOLA Configuration (Configuration 14) Lower delay, correct aliasing behavior. Complexity is very high.**



**Figure 41. Frequency Responses (Single Band-Pass and All-Pass) for the Specified WOLA Configuration (Configuration 15)**

Very low delay, but poor filter transitions. Aliasing is widely spread.

### Overlap-Add Configurations

Overlap-add are a special kind of configurations corresponding to  $N = La$ , and featuring particular aliasing properties. Usually, in such configurations, the Brennan window is not appropriate except if the synthesis filter is shorter than the analysis one, assuming a small block size  $R$  compared to  $La/2$  (like in Configuration 15). However, in such conditions, very smoothed gain settings should be

used. Actually, in most cases, both filters have the same lengths, and  $R$  is equal to  $La/2$ ,  $La/4$  or  $La/8$ . In such configurations, the filterbank process corresponds directly to a STDF. As mentioned in Chapter 2, another prototype filter can then be used at both analysis and synthesis, which is the sine lobe window or square root of Hann. Alternatively, one could use the Hann window at analysis, and a rectangular one at synthesis. In the hardware, as a

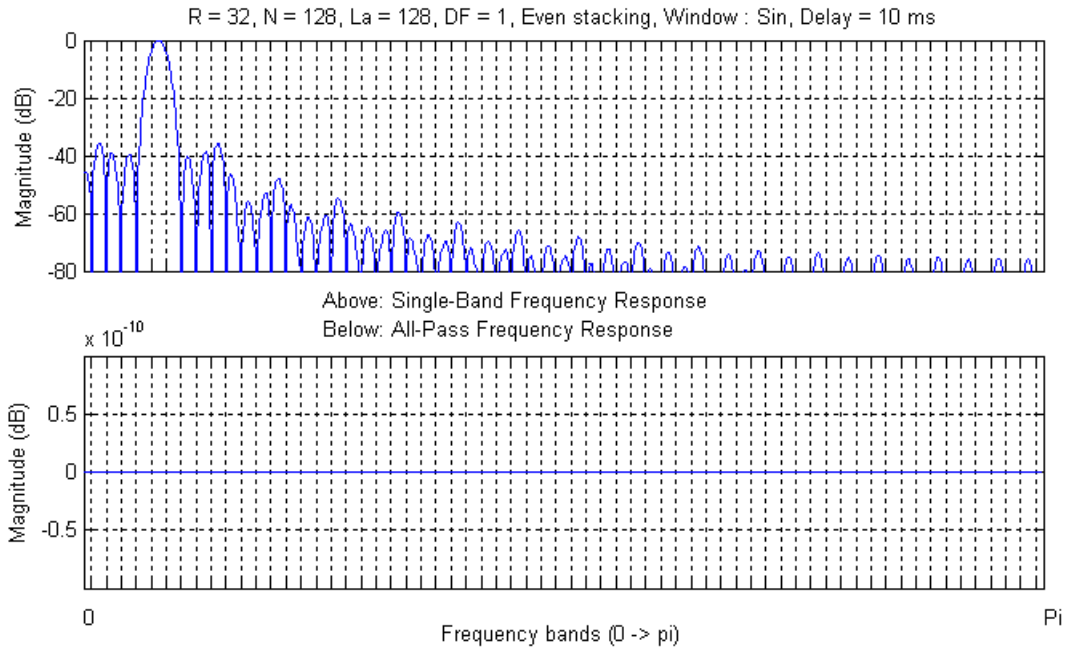
default, the sine lobe is always used when  $N = La$ , even in situations where  $Ls$  is shorter. It must be noted that the hardware uses a “symmetric – periodic” format for the default window. In such a format, the center of the symmetry occurs at index  $La/2$ :  $w(La/2) = 1$ . Then,  $w(La/2-1) = w(La/2+1)$ ,  $w(La/2-2) = w(La/2+2)$ , ..., which leads to  $w(0) = 0$  and  $w(La-1) = w(1)$ , which is different from 0. Using such a window allows for perfect reconstruction, in the absence of spectrum modification between analysis and synthesis.

Overlap–add configurations are often a good compromise in the SignaKlara architecture, when the required number of bands is high (64 or 128). This is because the filter length  $La$  cannot be four or eight times higher. The calculation load is

also modest, at least when the overlap is not high. However, one should be aware that such configurations are much more sensitive to gain applications than usual filterbank configurations. Gain shaping curves in the frequency should have a good amount of smoothing. Overlap–add configurations are really good choices in all types of applications that do not perform major modifications to the spectrum (like audio or speech coding, in which only quantization is performed in the frequency domain).

**Configuration 16**

Let’s have a look to a few overlap–add cases. In the first one (Figure 42), 64 bands are used; while the delay is 10ms. The overlap ratio is 3/4.



**Figure 42. Frequency Responses (Single Band–Pass and All–Pass) for the Specified WOLA Configuration (Configuration 16)**

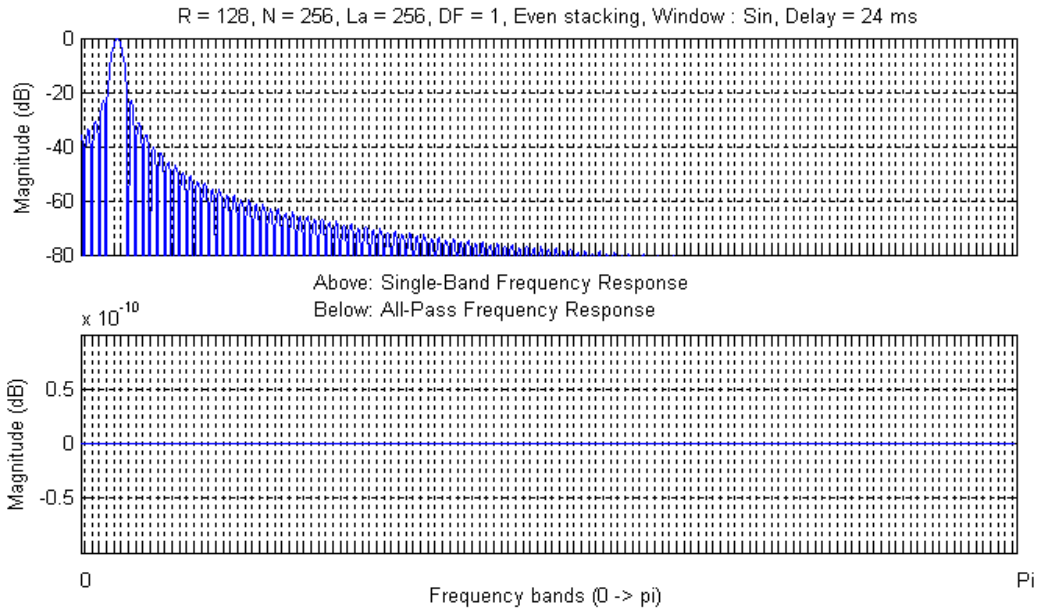
Such configurations are good compromises because of acceptable computational load, delay and aliasing, provided the gains applied in the frequency domain are not too sharp.

**Configuration 17**

In Configuration 17 (Figure 43), 128 bands are produced with a delay of 24 ms. Aliasing is totally cancelled on the

output when no gains are applied in the frequency domain, as it is typical in speech coding or recognition.

# AND8382/D

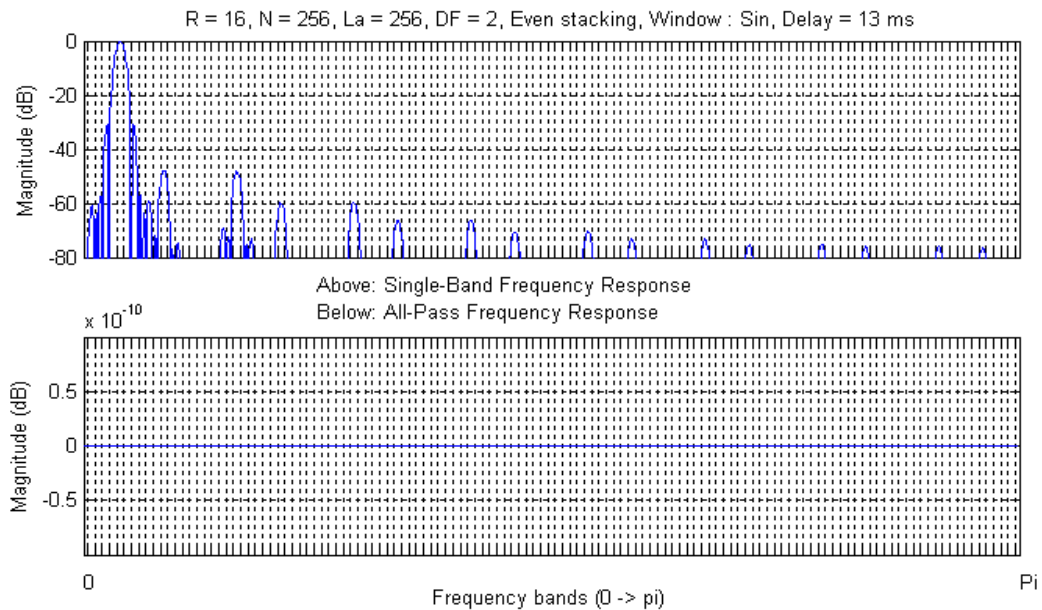


**Figure 43. Frequency Responses (Single Band-Pass and All-Pass) for the Specified WOLA Configuration (Configuration 17)**

This configuration corresponds to the short-term fourier transform with 50% overlap.

## Configuration 18

This configuration (Figure 44) shows an example with reduced delay (13 ms). Computational load is much higher, as a counterpart.



**Figure 44. Frequency Responses (Single Band-Pass and All-Pass) for the Specified WOLA Configuration (Configuration 18) Reduced Delay**

**Studying Relations Between STDFT and WOLA  
Overlap-add Configurations**

Before leaving this section, let's have a more detailed look to the relation between the STDFT and the WOLA filterbank approach, when using the latter in an overlap-add configuration. As stated above, those both approaches are the same. In reality, they are *nearly* the same. The following lines try to explain why this "nearly".

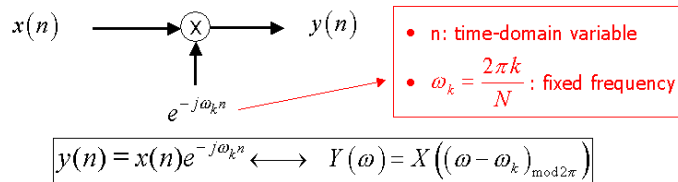
Let's consider the particular case in the STDFT, when the overlap is 50%:  $R = N/2$ . In order to implement the STDFT with the WOLA, the filter lengths should be  $L_a = L_s = N$ , choosing a Hann window at analysis and a rectangular window at synthesis, for example. Then, considering Figure 12 and 13, one can observe that the even stacking mode of the WOLA processing effectively corresponds to the STDFT. At analysis, the Hann window is applied first. Then, the "time-folding" additions of sub-blocks are by-passed as  $N = L_a$ . In the end, the FFT is performed. Similar observations can be done at synthesis. However, there is one difference: what about this circular shift, performed just before the FFT at analysis, and just after it, at synthesis?

The answer to that question requires us to come back to the complex-bandpass interpretation of the filterbank, in Figure 10 (for analysis), and understand what is this circular shift. In this figure, Step 3 of the processing is a complex multiplication of the channel signal  $\hat{X}_k(m)$  with the term  $\exp(-j\omega_k Rm)$ . In the complex bandpass interpretation, this complex multiplication is to be considered as applied to the

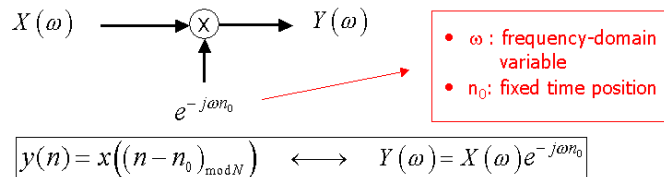
time-domain signal  $\hat{X}_k(m)$ ,  $m$  being the time-domain sample index and  $k$  specifying the particular channel. As a result a new time-domain signal is obtained,  $X_k(m)$ . As illustrated in the same figure, such a product in the time-domain corresponds to a shift of the  $k$ -th channel spectrum, in the frequency domain. This shift is performed by the amount  $-\omega_k R$ , and consequently, the channel is shifted at the origin of the frequency axis ( $\omega_k R$  was the original position of the particular channel). It is to be noted that  $m$  is the time-domain variable in the phasor  $\exp(-j\omega_k Rm)$ , while  $-\omega_k R$  represents the frequency domain shift.

What about changing now the way one considers the output of the analysis filterbank, thinking about it as a frequency domain signal,  $X_k(m)$  obtained at block  $m$  for all the channels indexed  $k$ ? Then, both signals  $X_k(m)$  and  $\hat{X}_k(m)$  become spectra, and the product  $X_k(m) = \exp(-j\omega_k Rm) \hat{X}_k(m)$  has now to be interpreted in the frequency domain. As a consequence, a new question can be stated: what is the product of a frequency domain signal  $\hat{X}_k(m)$  with the phasor  $\exp(-j\omega_k Rm)$ ? According to the time-frequency duality concepts of the signal processing theory [Pro96], it appears this time, that such an operation is a shift in the time-domain, more concretely, a shift by  $-Rm$  samples modulo  $N$ . It should be noted that from this point of view,  $\omega_k$  now plays the role of the frequency-domain variable. Both dual points of view are summarized in Figure 45.

- Modulation of a time-domain signal  $x(n)$  by  $\exp(-j\omega_k n) =$   
= Shift of the spectrum by  $-\omega_k$  modulo  $2\pi$ , in the frequency domain



- Modulation of a frequency-domain signal  $X(\omega)$  by  $\exp(-j\omega n_0) =$   
= Shift by  $-n_0$  samples modulo  $N$  in the time domain  
( $N$ =number of channels in frequency)



**Figure 45. Comparison of Complex-modulation Operations in the Time and Frequency Domains**

The WOLA filterbank actually exploits this duality: it considers the signal  $X_k(m)$  as a spectrum. Then, instead of performing the complex-modulation  $X_k(m) = \exp(-j\omega_k Rm) \hat{X}_k(m)$  as a product in the frequency domain (that is as a last

step, after the FFT), it performs it as a shift in the time-domain, that is before the FFT block. Applying a circular shift of the time-domain samples by  $-Rm$  modulo



$N$ , in the input buffer of FFT is actually much more efficient than performing a complex product, after the FFT.

As a consequence, the circular shift in the WOLA corresponds to Step 3 in *that* is it shifts the channel spectrum at the origin of the frequency axis. This operation is not performed in the STDFT approach. One can then wonder why such an operation in the WOLA approach? Let's try and further understand this operation, which actually concerns only the phase of the spectrum, and not its amplitude, making the spectrum of the channels to have *slowly varying* phase. In the 50% overlap configuration, let's have a look to the influence of that operation, when considered as a circular time-domain shift (a right shift !) in the input FFT buffer:

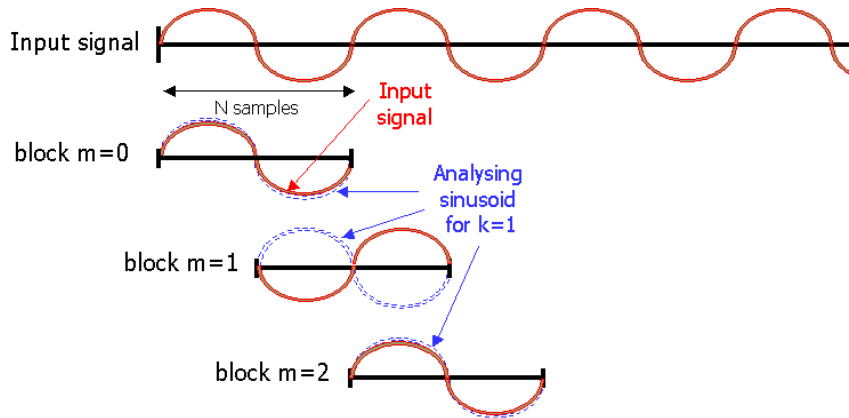
- In block  $m=0$ , the shift to apply is  $-Rm \bmod N = 0$  samples
- In block  $m=1$ , the shift to apply is  $-Rm \bmod N = -R = -N/2$  samples
- In block  $m=2$ , the shift to apply is  $-Rm \bmod N = -2R \bmod N = -N \bmod N = 0$  samples
- In block  $m=3$ , the shift to apply is  $-Rm \bmod N = -3R \bmod N = -3N/2 \bmod N = -N/2$  samples
- ...

This means that the input buffer is alternatively either untouched or its both halves are commuted. In order to interpret what happens, let's remember what is actually

performed when analyzing an input signal with an  $N$ -point DFT. In an FFT,

- Analyzing band  $k=0$  means trying to find a match between the input signal and a sinusoid having period 0 samples (that is DC).
- Analyzing band  $k=1$  means trying to find a match between the input signal and a sinusoid having period  $N$  samples, positioned at the center frequency  $\omega_1 = 2\pi/N$  of channel  $k=1$ .
- Analyzing band  $k=2$  means trying to find a match between the input signal and a sinusoid having period  $N/2$  samples, positioned at the center frequency  $\omega_2 = 4\pi/N$  of channel  $k=2$ .
- ...
- Analyzing band  $k$  means trying to find a match between the input signal and a sinusoid having period  $N/k$  samples, positioned at the center frequency  $\omega_k = 2\pi k/N$ .

As an example, let's assume that the input signal to be analyzed by the STDFT is precisely such a sine wave, centered at frequency  $\omega_1 = 2\pi/N$  (assuming  $k=1$ ). This input sinusoid has period  $N$ . The STDFT would perform the block-by-block analysis in the following way, as illustrated in .



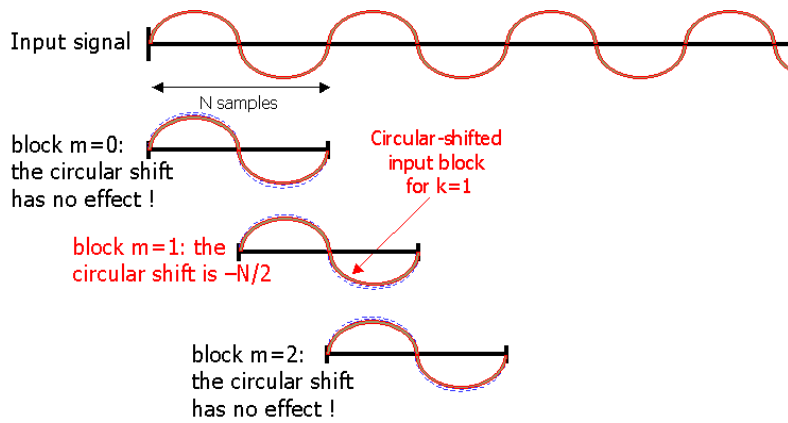
**Figure 46. STDFT Process: Analyzing an Input Sine Wave of Period  $N$  Samples**

The analyzing sinusoid in band  $k=1$  is illustrated in dotted blue, while the input signal is in red. Both signals are not in phase, in odd blocks.

One can observe, in block  $m = 1$ , that the analyzing sinusoid is not in phase with the input signal. Actually, this is true every other odd blocks  $m$ . Consequently, the phase of the spectrum will continuously jump between 0 and  $\pi$ , from

one block to the other. On the opposite, when the circular shift is performed, like in the WOLA realization, then the analyzing sinusoids are always in phase with the on-going input sine, as illustrated in Figure 47.





**Figure 47. Overlap-add Process with the WOLA: Analyzing an Input Sine Wave of Period  $N$  Samples**

The analyzing sinusoid in band  $k=1$  is illustrated in dotted blue, while the input signal is in red. Both signals are always in phase

Such a property is an interesting advantage in some particular applications, for example in order to generate tones with the WOLA, from the frequency domain. Effectively, in such an application, the phase of the synthesized tones has to stay coherent from one block to another. This is the reason why this phase alignment is included in the filterbank approach, appearing as well in the WOLA overlap-add configurations. In such cases, the phase of the frequency spectrum is slightly different than the STDFT one. However, this is only a question of phase interpretation. One just has to know which interpretation is used, in order to apply the correct processing when modifying the phase spectrum is required. There is actually no “good way” or “bad way”. Both approaches are just different ways, either of them being best appropriate in different situations.

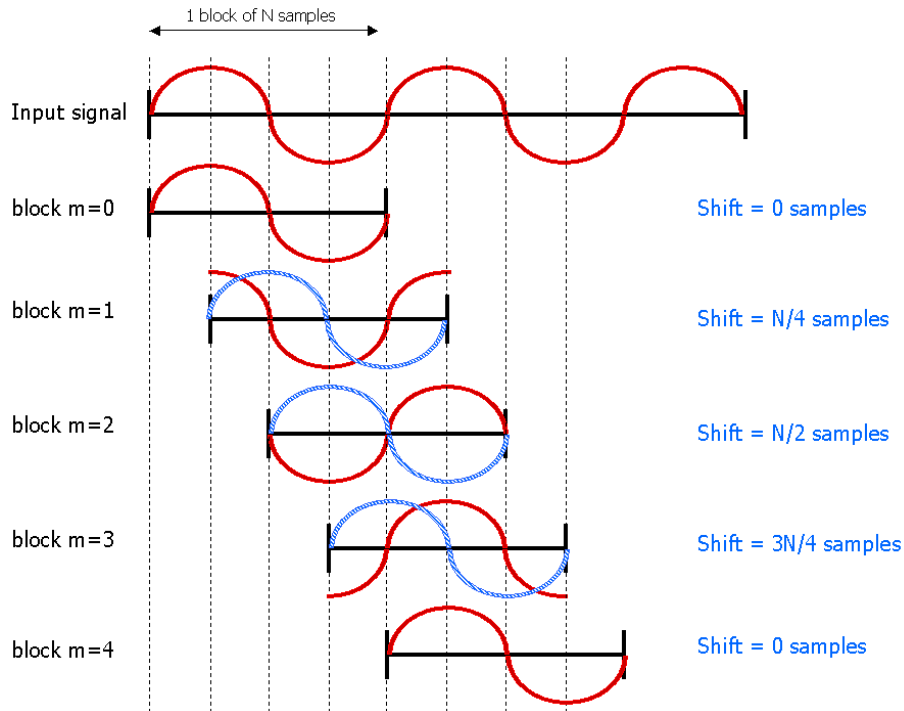
As a last observation, let’s mention that the shift depends on the relation between  $R$  and  $N$ . For example, when overlap is 50% ( $R = N/2$ ), the shift is annealed in all even blocks  $m$ , but also in all even indexed bands ( $k=0,2, 4, \dots$ ). Furthermore, the behavior is more complex when the overlap increases. The situation for  $R = N/4$  is illustrated in

Figure 48. On this figure, one can observe that the shifted signal is always in phase with the analyzing sine waves.

### Performing FFTs with the WOLA Filterbank

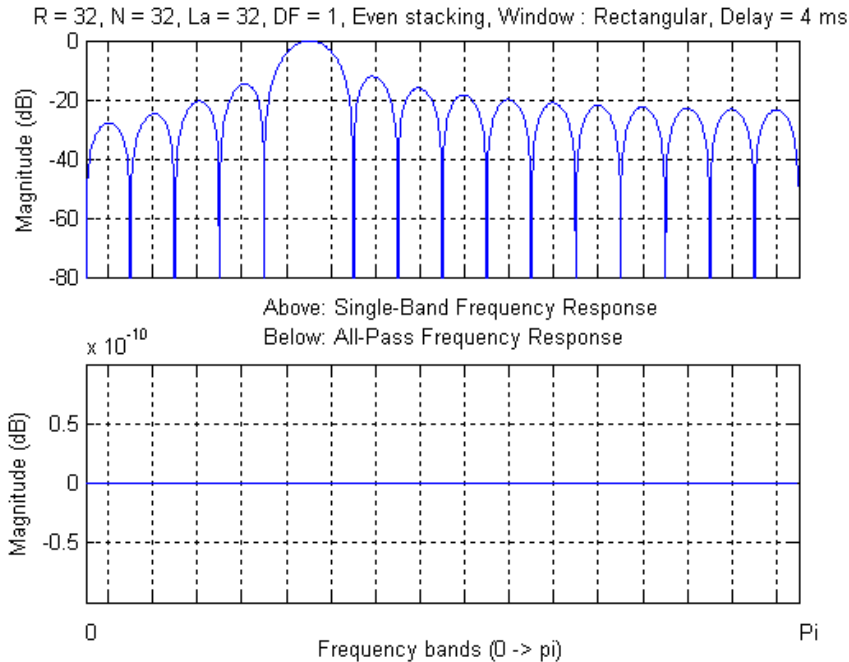
Looking at the WOLA structures in Figure 12 and 13, one can observe that it should be possible to perform an  $N$ -point FFT, respectively IFFT, with the WOLA, as FFT or IFFT blocks are included in the structures. In order to do that, however, the WOLA should be configured in a special way, having all other functional blocks (sign sequencing, windowing, time-folding, shifts) to have no influence during the chain process. In this goal, the following configuration must be selected:  $N = R = La = Ls$ . Furthermore, the even stacking mode should be selected, and rectangular windows should be used at both analysis and synthesis. Such configurations are not part of the microcode set, but they can be provided on demand.

It is interesting to observe that the FFT and inverse FFT process, performed in successive non-overlapping blocks ( $R=N=La=Ls$ ) do provide perfect reconstruction. However, the frequency domain behavior is quite bad, as can be observed on Figure 49, when  $N = 32$ . Applying frequency shaping before IFFT would not be possible.



**Figure 48. WOLA Overlap-add Process: Illustration of the Shift Process When  $R=N/4$ , in the Case of an Input Sine Wave of Period  $N$**

The input signal is in red, while the shifted signal is represented in patterned blue. The shifted signal is always in phase with the analyzing sinusoid (not represented on the Figure).



**Figure 49. Successive FFTs and IFFTs Processes, Using Non-overlapping Blocks (Rectangular Windows)**

Perfect reconstruction is ensured in the time-domain. However, very high aliasing level is observed in the frequency domain, preventing any gain application to be performed.

**Using Special Microcodes**

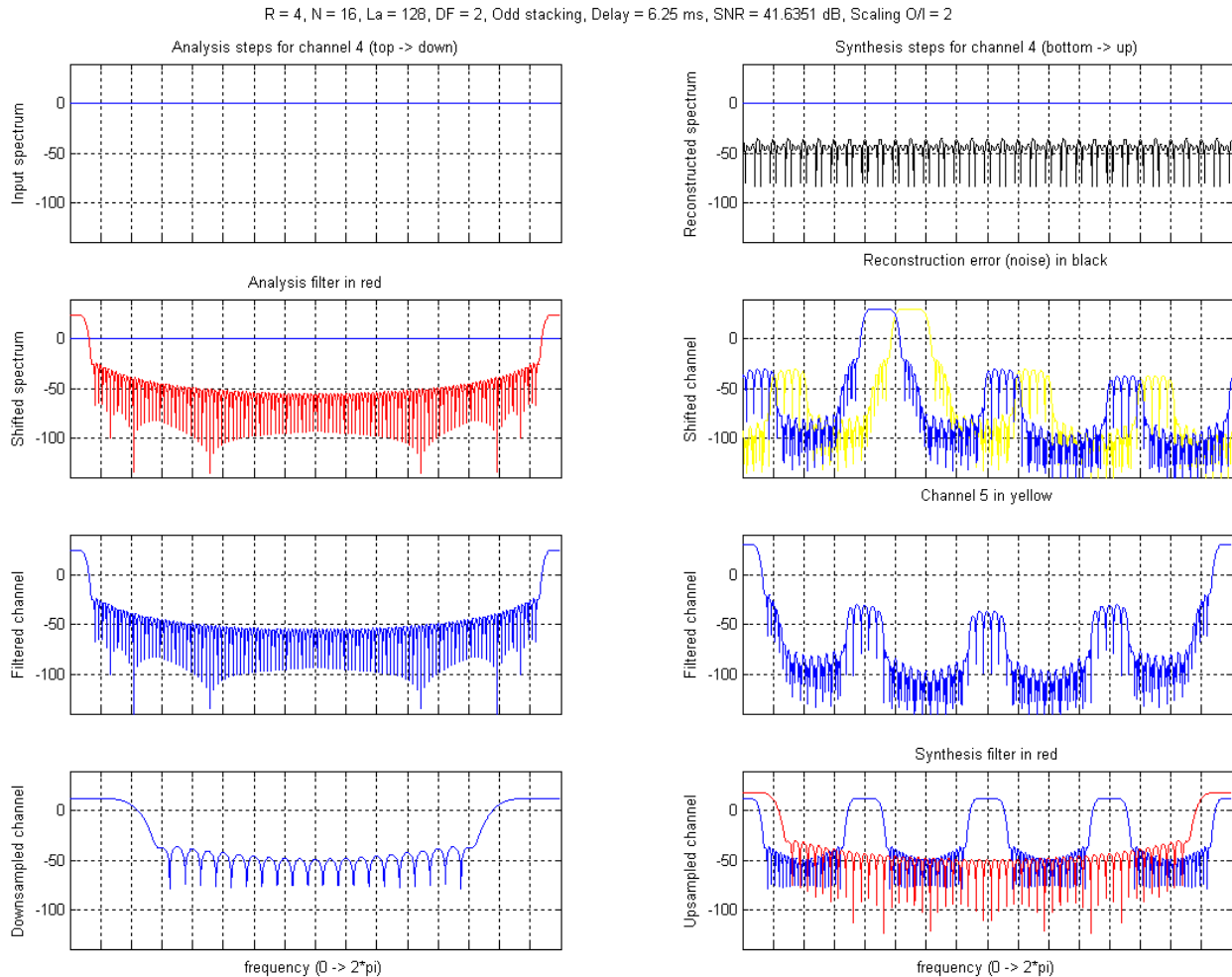
Special WOLA configurations can sometimes be designed and made available to customers. Special requests should be addressed to ON Semiconductor’s support team in order to discuss their feasibility. Only the most used configurations are included in ON Semiconductor’s evaluation and development kit.

**REFERENCES**

[Cro83] R.E. Crochiere, L.R. Rabiner, “Multirate Digital Signal Processing”, Prentice Hall, 1983.  
 [Vai93] P.P. Vaidyanathan, “Multirate Systems and Filter Banks”, Prentice Hall, 1993.  
 [Pro96] J.G. Proakis, D.G. Manolakis, “Digital Signal Processing, Principles, Algorithms, and Applications”, Prentice Hall, 1996.

**ANNEX 1**

According to Figures 6 and 7, this annex shows the different signal spectra, at every step in the complex-modulation filterbank (original interpretation). The first set of plots shows the situation in one particular channel, for an impulse input signal. The second set of plots involves a white noise, while a narrow-band white noise is used in the third and fourth sets. Finally, a sine wave is shown on the last plots. In all cases, the same WOLA configuration is used, and Channel 4 is illustrated. The filter is the default Brennan one. The analysis steps are shown on the left, starting on top. The synthesis steps appear on the right, from the bottom. The scale of the different signals is as in the WOLA filterbank process.



**Figure 50. Analysis and Synthesis Steps in the Complex-modulation Filterbank (Original Interpretation) Impulse Signal**

R = 4, N = 16, La = 128, DF = 2, Odd stacking, Delay = 6.25 ms, SNR = 41.9795 dB, Scaling O/I = 2

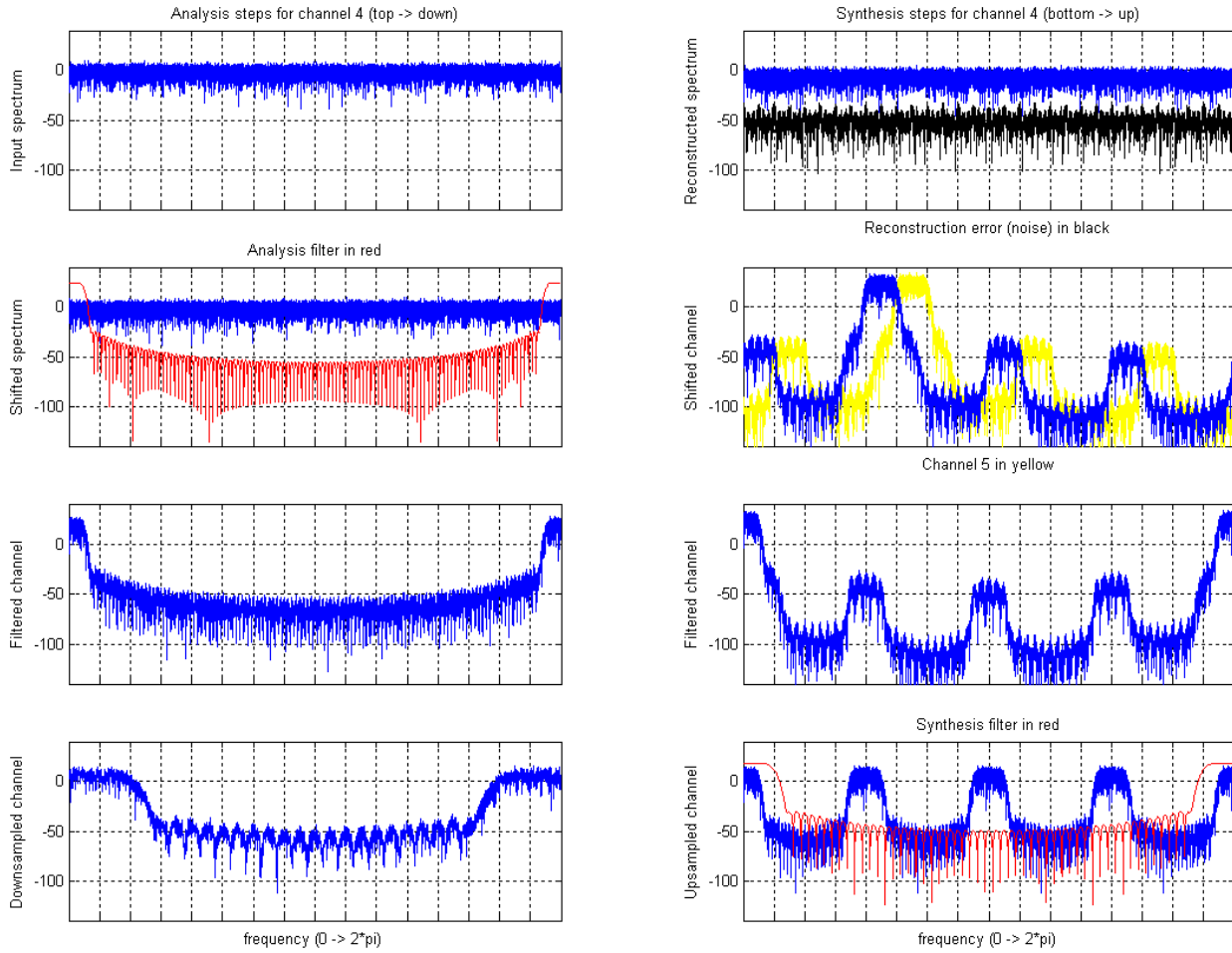
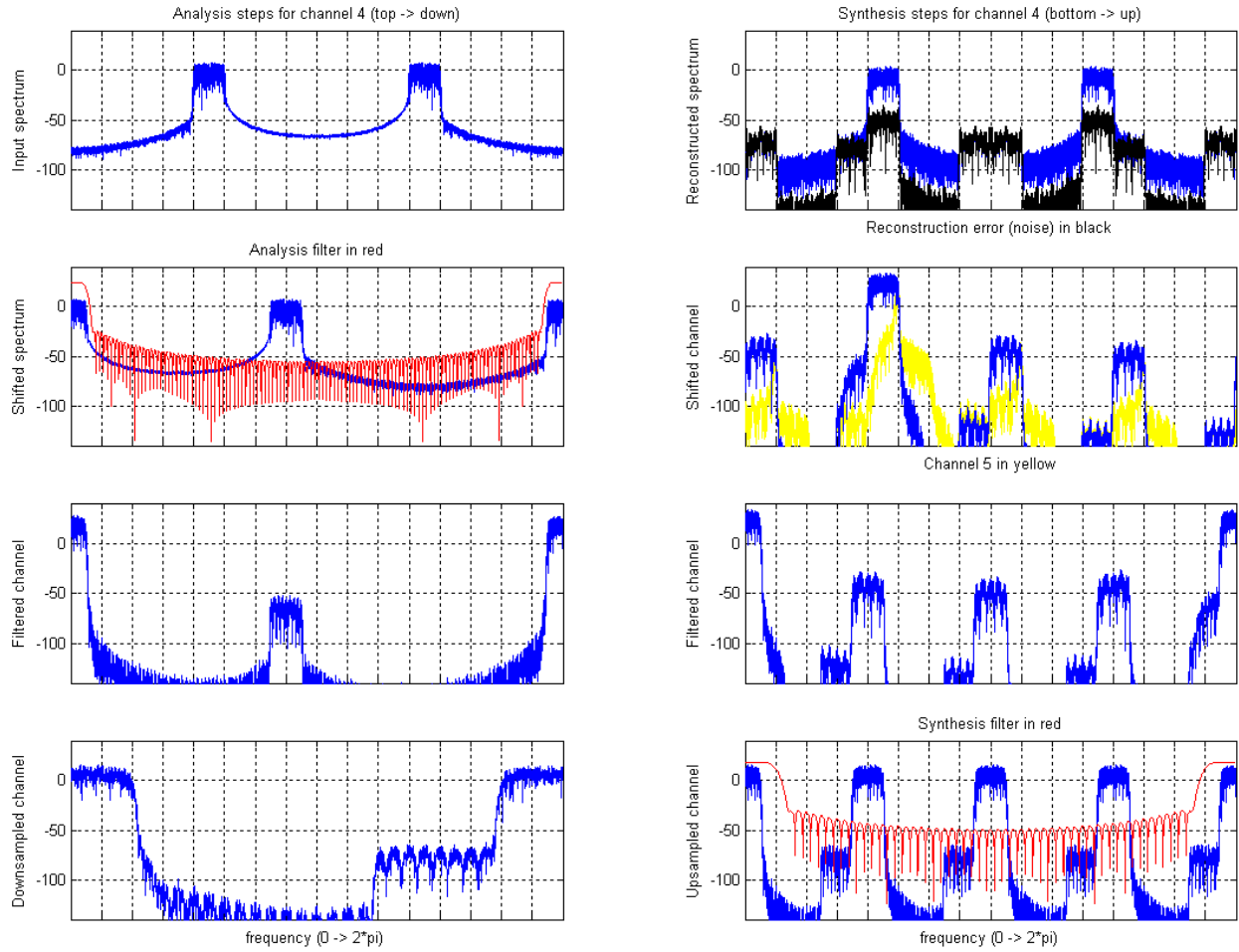


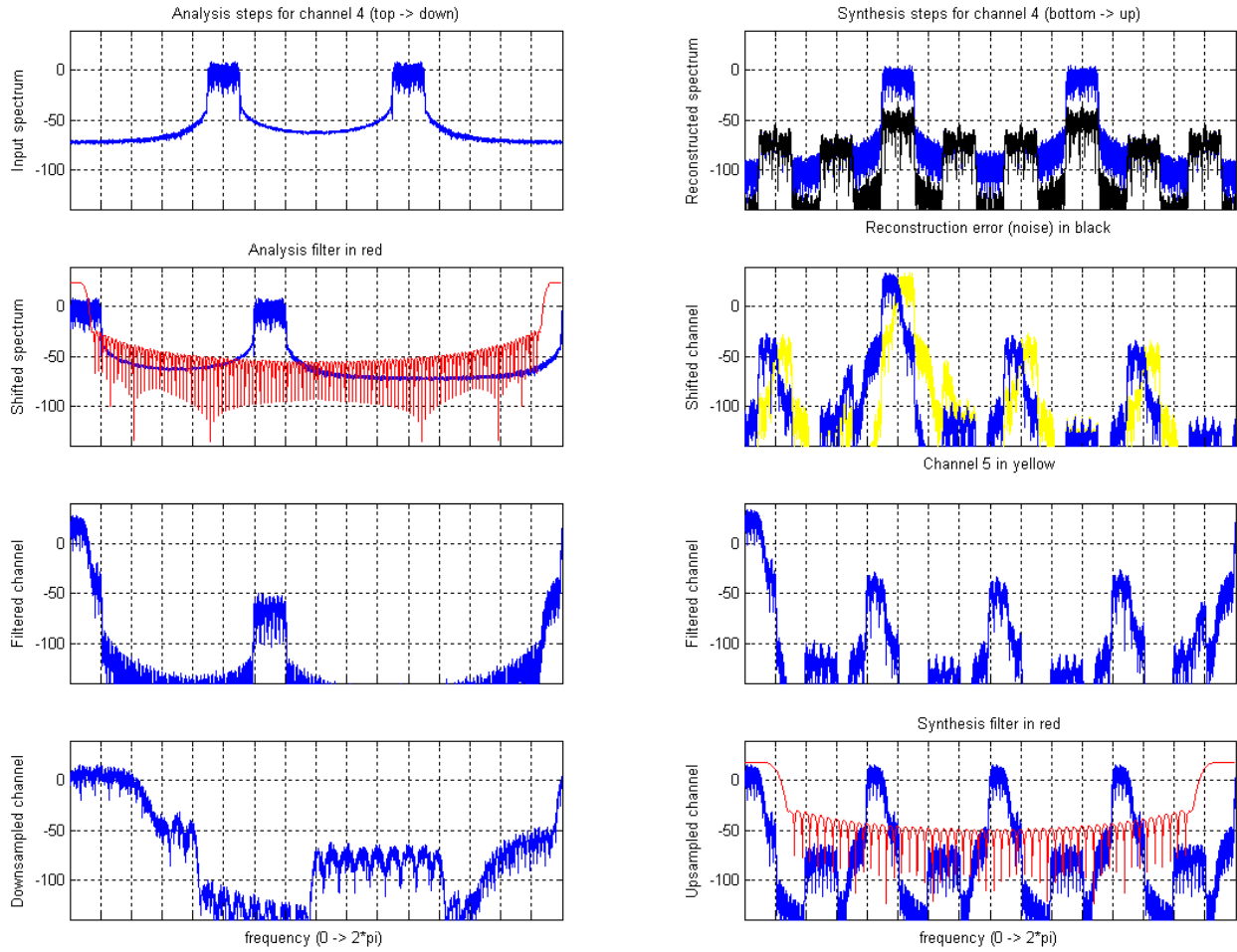
Figure 51. Analysis and Synthesis Steps in the Complex-modulation Filterbank (Original Interpretation) White Noise Signal

R = 4, N = 16, La = 128, DF = 2, Odd stacking, Delay = 6.25 ms, SNR = 42.2299 dB, Scaling O/I = 2

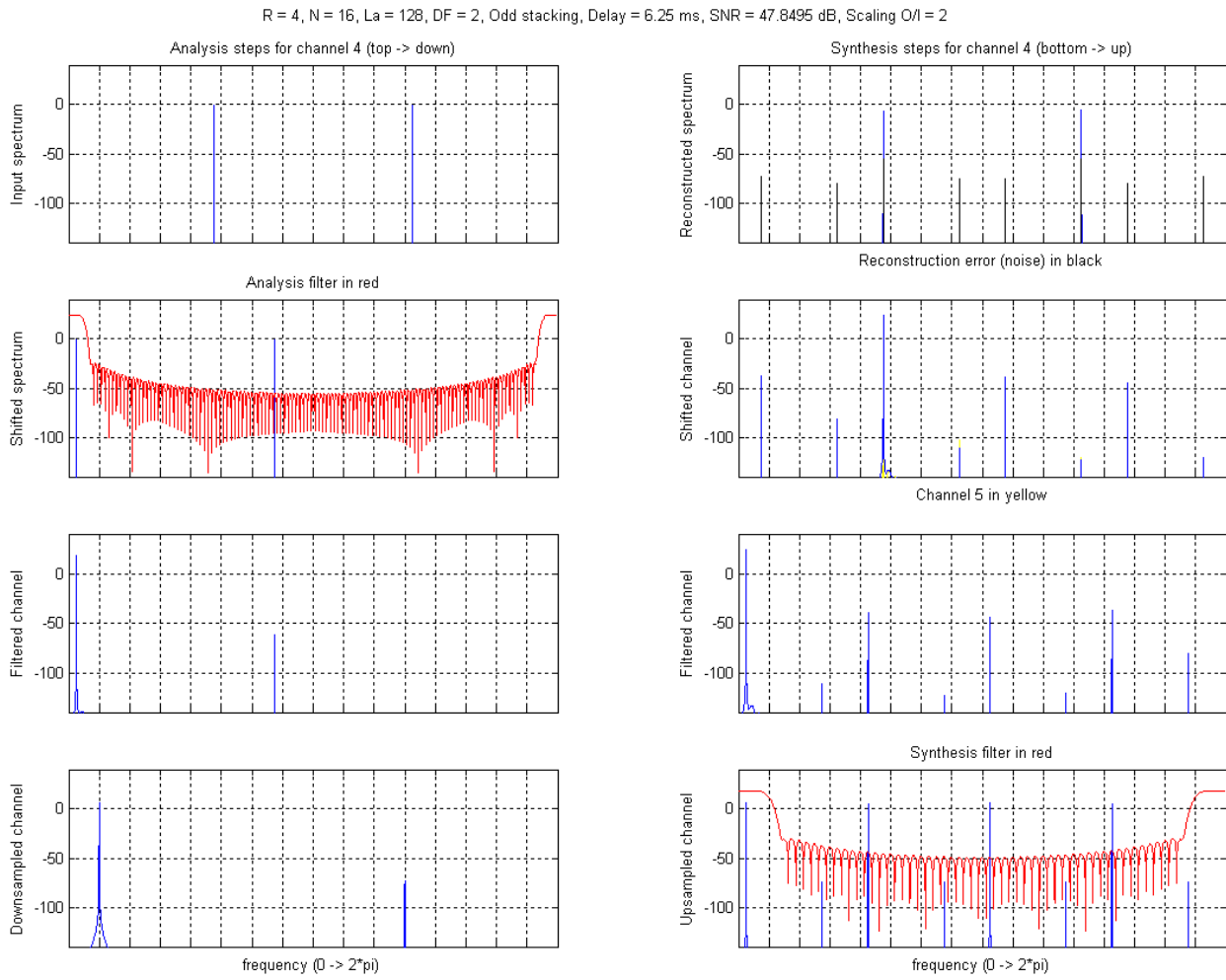


**Figure 52. Analysis and Synthesis Steps in the Complex-modulation Filterbank (Original Interpretation) Narrow-band white noise signal in Channel 4**

R = 4, N = 16, La = 128, DF = 2, Odd stacking, Delay = 6.25 ms, SNR = 42.4059 dB, Scaling O/I = 2



**Figure 53. Analysis and Synthesis Steps in the Complex-modulation Filterbank (Original Interpretation) Narrow-band white noise signal C7003 Centered on the Transition Between Channels 4 and 5**



**Figure 54. Analysis and Synthesis Steps in the Complex-modulation Filterbank (original interpretation) Sine Wave**

ANNEX 2

According to Figures 10 and 11, this annex shows the different signal spectra, at every step in the complex-modulation filterbank. This time, the complex bandpass interpretation is considered. The first set of plots shows the situation in one particular channel, for an impulse input signal. The second set of plots involves a white noise, while a narrow-band white noise is used in the third and

fourth sets. Finally, a sine wave is shown on the last plots. In all cases, the same WOLA configuration is used, and Channel 4 is illustrated. The filter is the default Brennan one. The analysis steps are shown on the left, starting on top. The synthesis steps appear on the right, from the bottom. The scale of the different signals is as in the WOLA filterbank process.

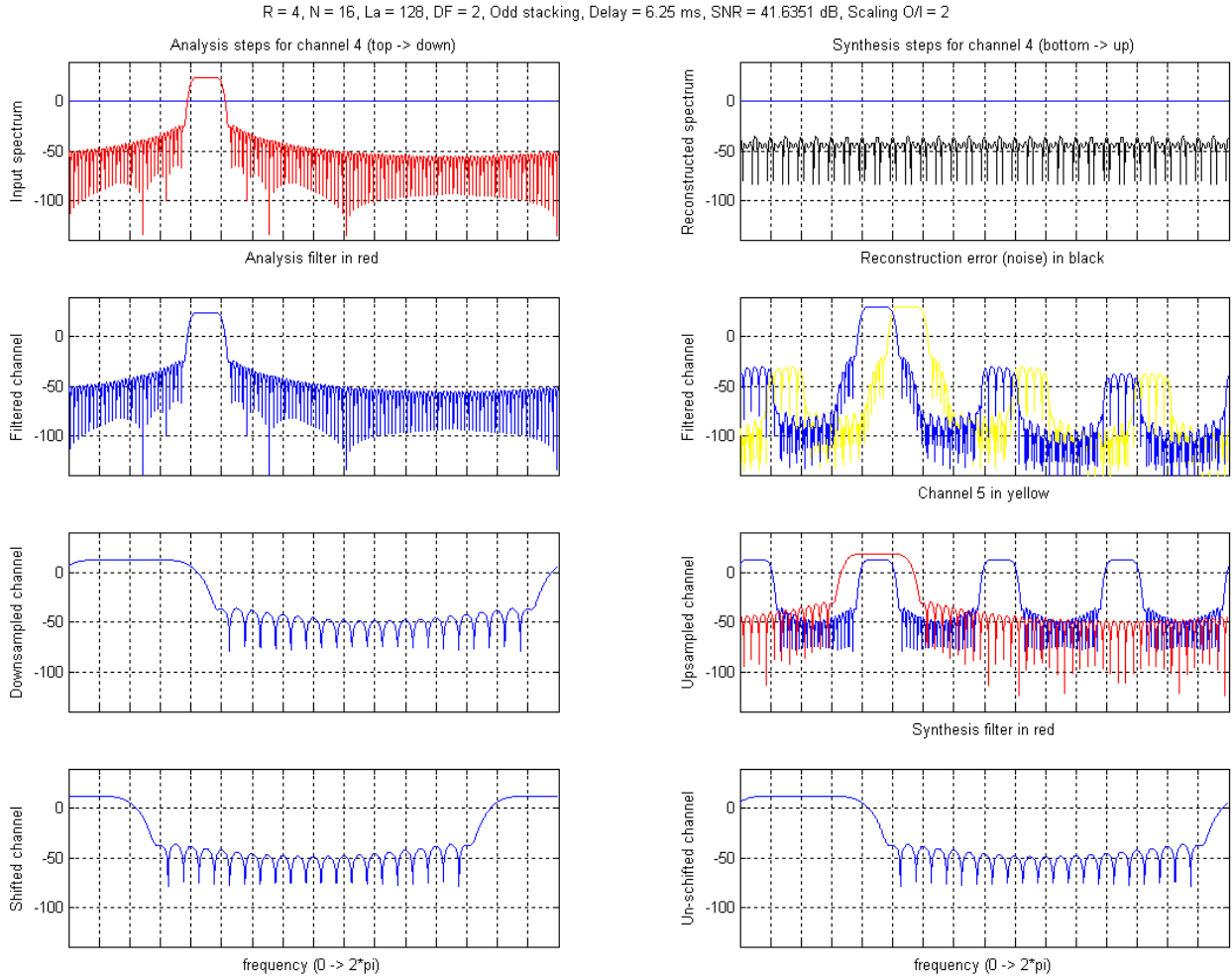


Figure 55. Analysis and Synthesis Steps in the Complex-modulation Filterbank (Complex Bandpass Interpretation) Impulse Signal



R = 4, N = 16, La = 128, DF = 2, Odd stacking, Delay = 6.25 ms, SNR = 41.9795 dB, Scaling O/I = 2

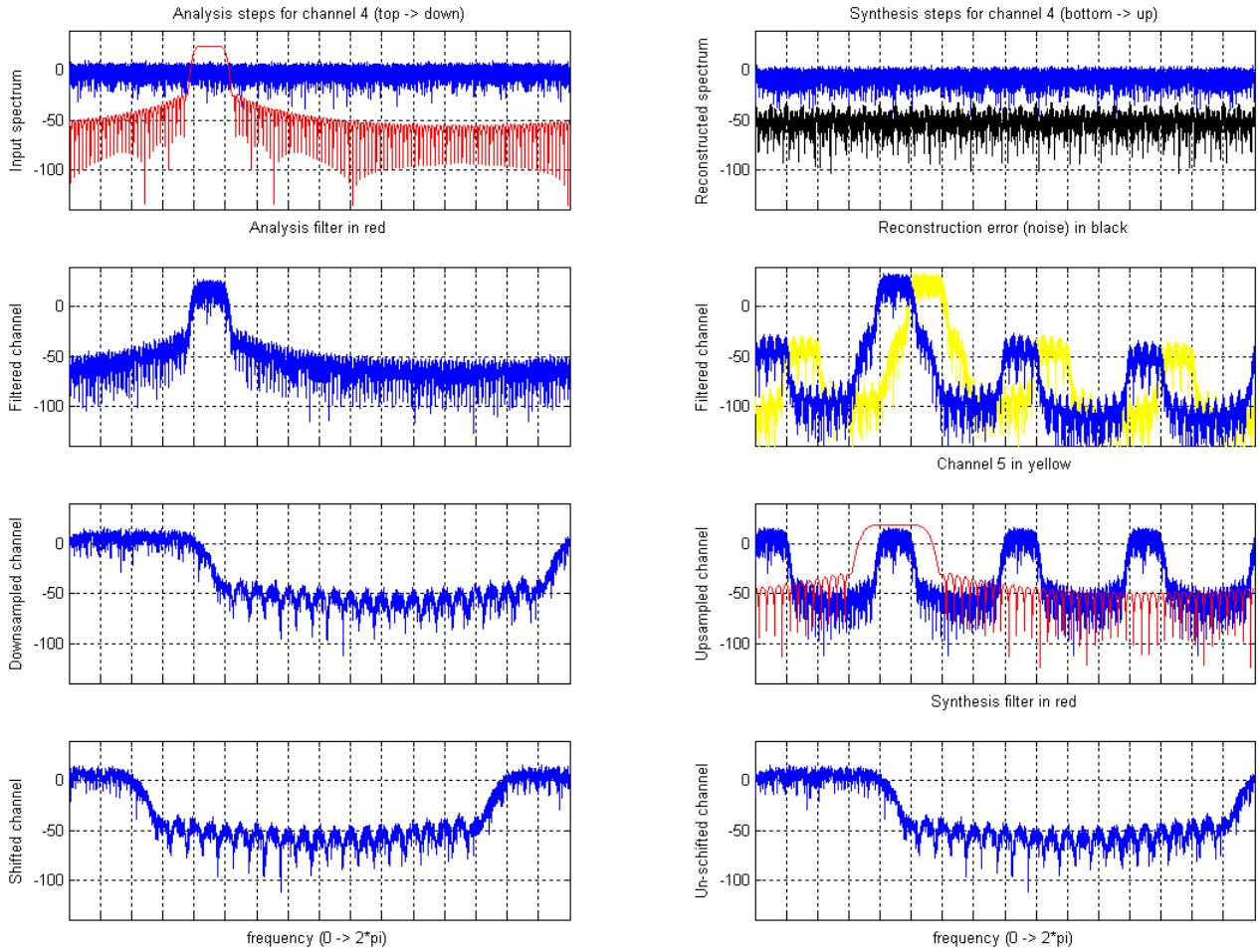
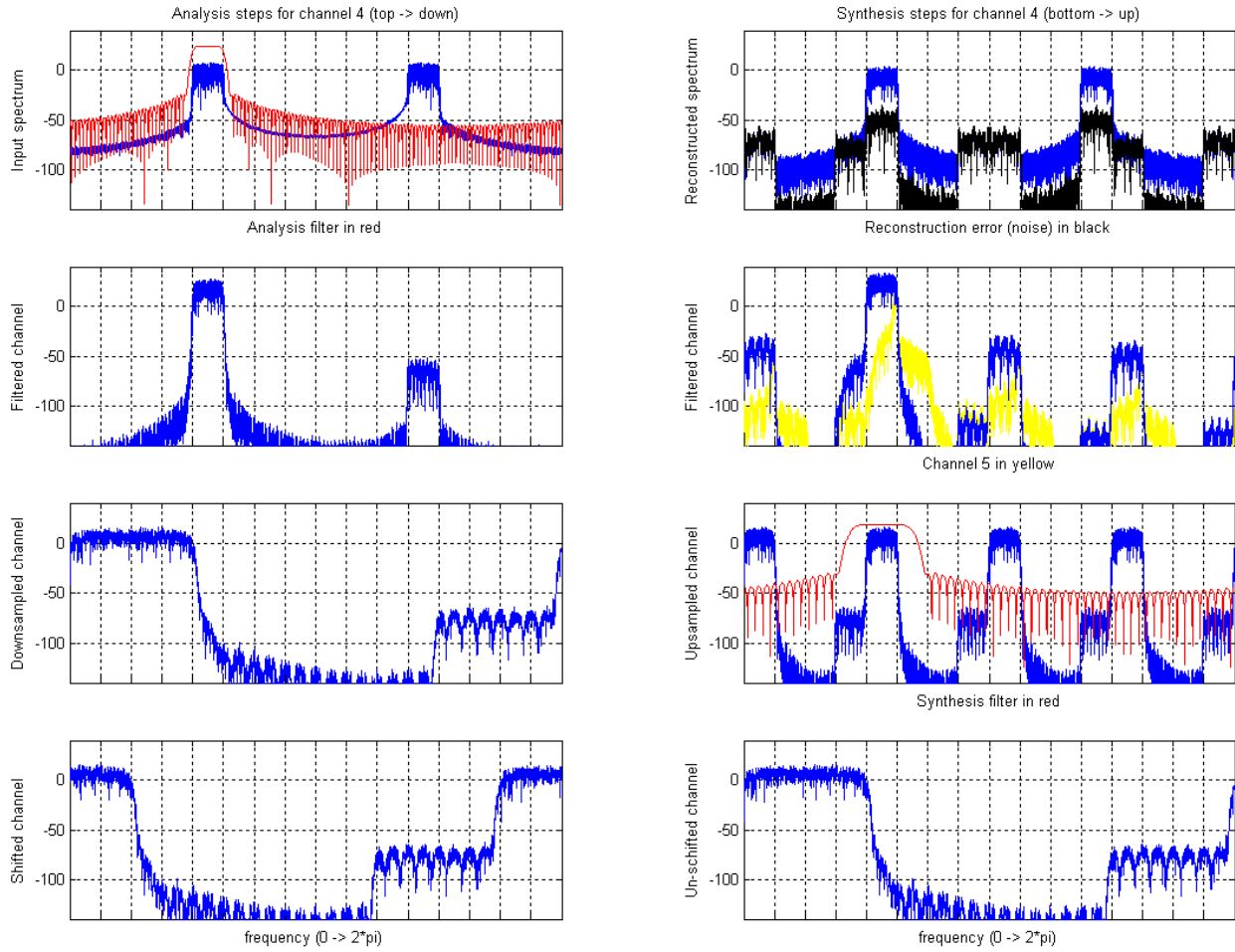


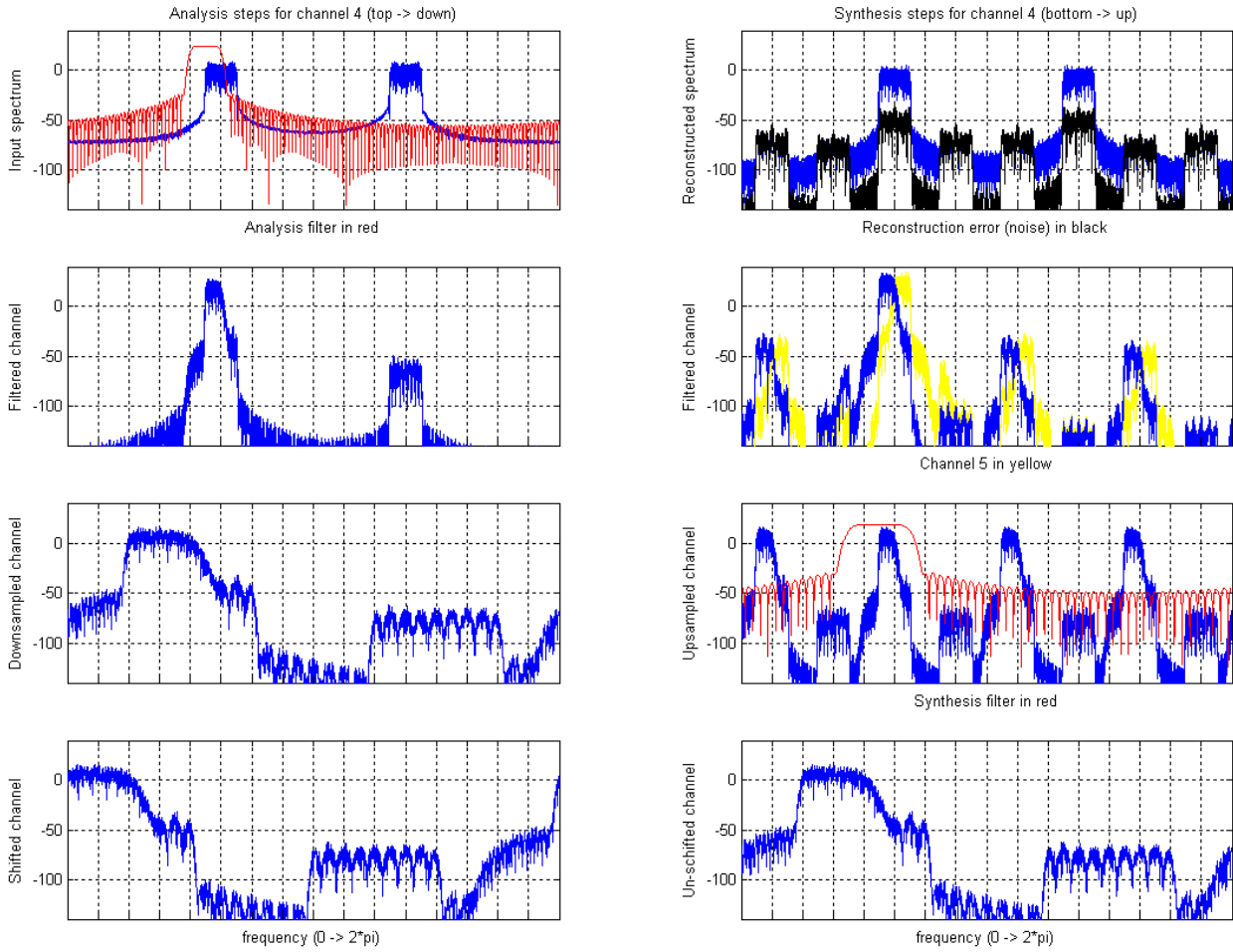
Figure 56. Analysis and Synthesis Steps in the Complex-modulation Filterbank (Complex Bandpass Interpretation) White Noise Signal

R = 4, N = 16, La = 128, DF = 2, Odd stacking, Delay = 6.25 ms, SNR = 42.2299 dB, Scaling O/I = 2



**Figure 57. Analysis and Synthesis Steps in the Complex-modulation Filterbank (Complex-Bandpass Interpretation) Narrow-Band White Noise Signal in Channel 4.**

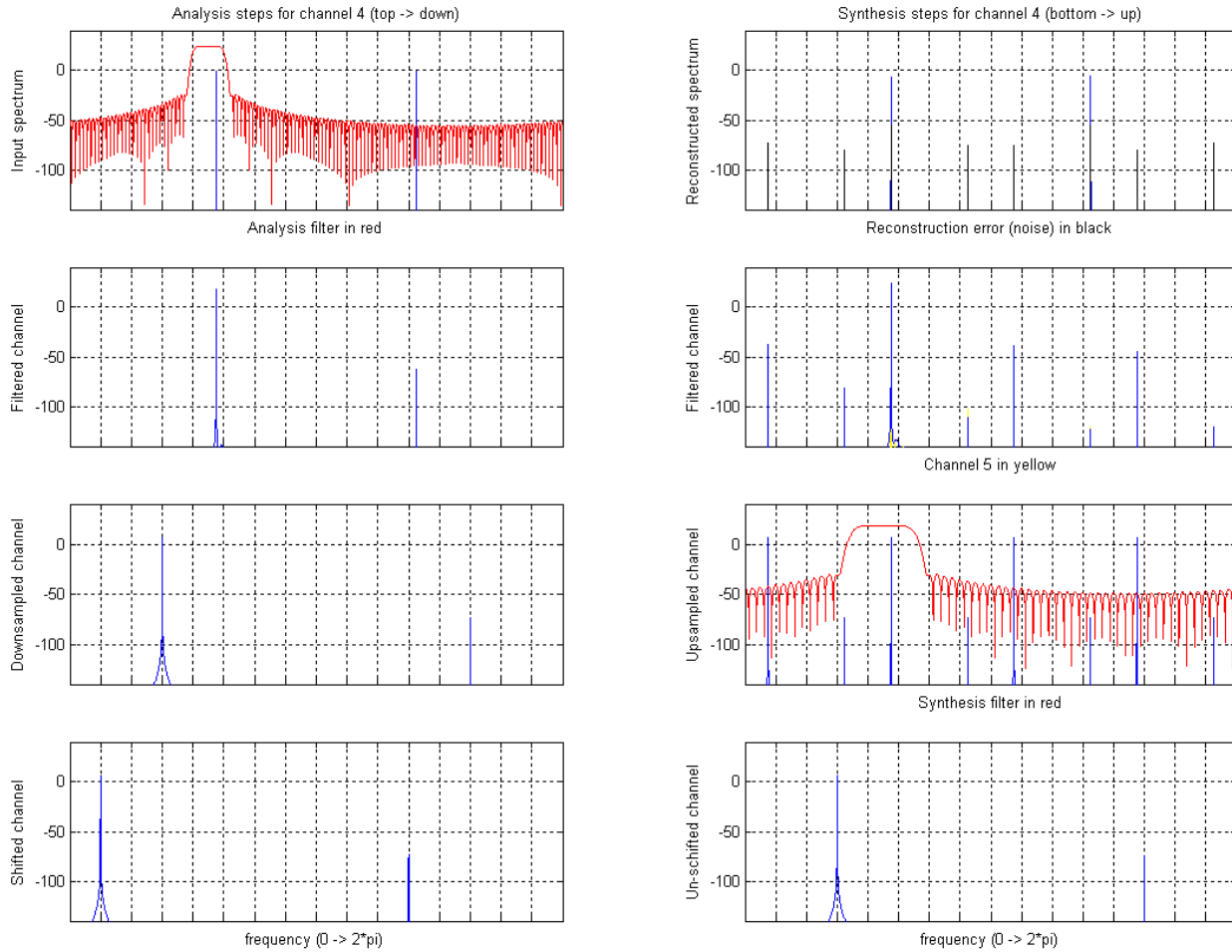
R = 4, N = 16, La = 128, DF = 2, Odd stacking, Delay = 6.25 ms, SNR = 42.4059 dB, Scaling O/I = 2



**Figure 58. Analysis and Synthesis Steps in the Complex-modulation Filterbank (Complex-Bandpass Interpretation) Narrow-Band White Noise Signal Centered on the Transition Between Channels 4 and 5.**


# AND8382/D

R = 4, N = 16, La = 128, DF = 2, Odd stacking, Delay = 6.25 ms, SNR = 47.8495 dB, Scaling O/I = 2



**Figure 59. Analysis and Synthesis Steps in the Complex-modulation Filterbank (Complex-Bandpass Interpretation) Sine Wave**

BelaSigna, Orela, and SignaKlara are registered trademarks of Semiconductor Components Industries, LLC (SCILLC).  
 Toccata Plus is a trademark of Semiconductor Components Industries, LLC (SCILLC).  
 MATLAB is a registered trademark of The MathWorks, Inc.

**ON Semiconductor** and  are registered trademarks of Semiconductor Components Industries, LLC (SCILLC). SCILLC reserves the right to make changes without further notice to any products herein. SCILLC makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does SCILLC assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation special, consequential or incidental damages. "Typical" parameters which may be provided in SCILLC data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. SCILLC does not convey any license under its patent rights nor the rights of others. SCILLC products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the SCILLC product could create a situation where personal injury or death may occur. Should Buyer purchase or use SCILLC products for any such unintended or unauthorized application, Buyer shall indemnify and hold SCILLC and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that SCILLC was negligent regarding the design or manufacture of the part. SCILLC is an Equal Opportunity/Affirmative Action Employer. This literature is subject to all applicable copyright laws and is not for resale in any manner.

## PUBLICATION ORDERING INFORMATION

**LITERATURE FULFILLMENT:**  
 Literature Distribution Center for ON Semiconductor  
 P.O. Box 5163, Denver, Colorado 80217 USA  
**Phone:** 303-675-2175 or 800-344-3860 Toll Free USA/Canada  
**Fax:** 303-675-2176 or 800-344-3867 Toll Free USA/Canada  
**Email:** orderlit@onsemi.com

**N. American Technical Support:** 800-282-9855 Toll Free  
 USA/Canada  
**Europe, Middle East and Africa Technical Support:**  
 Phone: 421 33 790 2910  
**Japan Customer Focus Center**  
 Phone: 81-3-5773-3850

**ON Semiconductor Website:** [www.onsemi.com](http://www.onsemi.com)  
**Order Literature:** <http://www.onsemi.com/orderlit>  
 For additional information, please contact your local  
 Sales Representative