



ON Semiconductor®

<http://onsemi.com>

Introduction to Audio Processing Using the WOLA Filterbank Coprocessor

APPLICATION NOTE

This application note is applicable to: Toccata Plus™, BelaSigna® 200, Orela® 4500 Series

INTRODUCTION

The Toccata Plus, BelaSigna 200 and Orela 4500 audio systems consist of a 16-bit programmable DSP and two audio coprocessors: the Input/Output Processor (IOP) and the WOLA filterbank coprocessor.

Both the IOP and the WOLA filterbank coprocessor are integrated into the Toccata Plus, BelaSigna 200 and Orela 4500 series products, as dedicated coprocessors, running in parallel with the main DSP core (R_{Core}). Both the IOP and the WOLA filterbank coprocessor are highly configurable. They are major components of the audio signal processing chain.

Specifically, the IOP automatically and without user intervention, manages the transfer of digital audio data from the input stage (ADC) to the input FIFO and from the output FIFO to the output stage (DAC). The WOLA coprocessor implements a weighted overlap-add filterbank that performs time domain-to-frequency domain and frequency domain-to-time domain transforms efficiently. Numeric precision is maintained by using block floating-point arithmetic in the WOLA filterbank coprocessor.

This document provides an overview of the features and typical usage of the IOP and the WOLA filterbank coprocessor. It also describes the basic considerations when selecting WOLA filterbank parameters and compromises for specific applications.

For more information regarding signal processing and theoretical aspects of the WOLA filterbank coprocessor, refer to the “WOLA Filterbank Coprocessor: Introductory Concepts and Techniques” tutorial, which provides more in-depth examples for developing on the DSP architecture. The goal of this document is to address the theoretical

aspects of the WOLA filterbank coprocessor and describe the influence of each involved parameter. This tutorial should be read prior to selecting a filterbank configuration for a particular application. Block floating-point aspects in the WOLA filterbank coprocessor are described in a dedicated application note “Using Block Floating-Point in the WOLA Filterbank Coprocessor”.

AUDIO PROCESSING MODES

The Toccata Plus, BelaSigna 200 and Orela 4500 chips are each a complete two-channel (stereo) system. Various audio processing modes are available, depending upon how many ADCs and DACs¹ are used.

The WOLA filterbank coprocessor can operate in one of three modes:

1. Single input, single output (i.e., mono)
2. Two inputs, single output (i.e., simple stereo)
3. Two inputs, two outputs (i.e., full stereo)

Furthermore, the time-domain digital audio data can be accessed before the WOLA signal processing is applied. Two time-domain input signals may be combined (by the code executed on the DSP) prior to WOLA signal processing. In this case, the WOLA filterbank coprocessor operates on a single input and generates a single output signal (i.e., digital mixed mode). Finally, the time-domain digital audio data can be transferred directly from the input FIFO to the output FIFO, bypassing the WOLA filterbank coprocessor completely.

The mode of operation is determined as part of configuring the IOP and WOLA filterbank coprocessor. This is a basic consideration when designing an audio signal processing application. During program execution, the mode of operation generally should not be changed on the fly.

1. For purposes of this document “DAC” will refer to either the D/A converter or the direct digital output.

IOP OPERATION

Once configured and started, the IOP continuously operates in the background. The IOP manages three tasks:

1. Accumulates the next block of audio data from the input stage in the input FIFO,
2. Outputs the previous block of audio data from the output FIFO to the output stage and
3. Coordinates utilization of the current block of audio data by the WOLA filterbank coprocessor.

The IOP has a built-in mechanism for muting the output. To enable this feature, set the IOP_AUTOMUTE_ENABLE bit in the D_IOP_CONFIGURE register. When enabled, the IOP will output zeros instead of the audio data if a complete block of audio data has not been written to the output FIFO within the block frame.

WOLA OPERATION

The WOLA filterbank coprocessor design is based on the complex-modulation approach and is efficiently implemented using the weighted overlap-add structure². The filterbank can be used to transform the time-domain input signal (mono or stereo) into several uniform frequency bands, working either on a sample-by-sample or a block-by-block basis. In the frequency domain, modifications can be applied to the spectrum (changing magnitude and phase), and then, the time-domain output signal can be reconstructed to facilitate these transformations. The WOLA filterbank coprocessor can perform the following functions:

- **Analysis:** Decomposes the input signal into several frequency bands.
- **Gain Application:** Multiplies the frequency-domain data with real or complex gains.
- **Synthesis:** Performs frequency-to-time transformation.

The audio characteristics of the WOLA filterbank coprocessor are determined by a set of configurable parameters. Parameter selection can be tuned for a number of filterbank audio bands and bandwidth, group delay and audio performance/aliasing distortion.

The WOLA filterbank coprocessor is not a simple FFT/IFFT coprocessor. It is a filterbank. While the WOLA filterbank coprocessor can be configured to perform a Fast Fourier Transform (FFT) and Inverse Fast Fourier Transform (IFFT), the WOLA architecture allows for distortion, delay and frequency/time resolution trade-offs, as explained in the following sections.

WOLA Audio Modes and Gain Application

The WOLA filterbank coprocessor can operate on one or two audio channels, and the following three modes are possible:

- **Mono:** Only one audio channel is processed; all three WOLA filterbank coprocessor operations can be applied (analysis, gain application and synthesis).

- **Simple Stereo:** Two input audio channels are processed by the WOLA analysis and gain application operations. Different gains can be applied to each input channel. After gain application, averaging or mixing is used to combine the data in both channels. The result after the WOLA synthesis operation is one output channel.
- **Full Stereo:** Two independent audio signal channels are processed through all three WOLA operations. Different gains can be applied to each input channel.

WOLA PARAMETERS

The filters, which are used during analysis (decomposition of the input signal into several frequency bands) and synthesis (reconstruction), are a key determinant for the performance of the WOLA filterbank coprocessor. Selection of appropriate filters is crucial in order to satisfy the fundamental signal processing constraints of the filterbank. Unfortunately, the choice of proper filters is not always straightforward. For that reason, default filter shapes (finite impulse responses, sometimes also called “windows”) are provided. While filter customization is possible, it is recommended that the default filters be used as they provide a very satisfactory behavior in most configurations.

Once the analysis and synthesis filter impulse responses are set, the following parameters can be selected for controlling the filterbank configuration:

- Number of frequency bands (N/2)
- Input block size (R) and oversampling factor (OS)
- Analysis filter length (La)
- Synthesis filter length (Ls)
- Band stacking (even or odd)
- Audio mode (mono, simple stereo or full stereo)
- Gain mode (real or complex)

The choice of WOLA filterbank parameters has consequences for the amount of aliasing (output audio quality), group delay, calculation load, time resolution, and frequency resolution. Proper selection of these parameters is important to obtain the best performance for the intended application. ON Semiconductor provides simulation tools (either in MATLAB® and Simulink® environments or as C-language functions), to assist in the selection of the filterbank parameters. The “WOLA Filterbank Coprocessor: Introductory Concepts and Techniques” tutorial is also available for reference. Using the simulation tools is highly recommended as they include a floating-point model of the WOLA filterbank coprocessor. The simulation tools are also useful when mapping a previously designed algorithm onto the WOLA filterbank architecture.

2. Refer to the “WOLA Filterbank Coprocessor: Introductory Concepts And Techniques” tutorial for more information.

The following sections describe the effect of various WOLA parameters on the performance characteristics.

Number of Frequency Bands (N/2)

The number of audio bands is determined by the parameter N, which is defined as the number of frequency bins from $f = 0$ to the sampling frequency $f = f_s$. Consequently, assuming real input signals and considering only the non-redundant part of the spectrum (from $f = 0$ to the Nyquist frequency $f = f_s/2$), the number of bands is $N/2$ (odd stacking situation) or $N/2 + 1$ (even stacking, see below). When the filterbank is configured to perform a single FFT/IFFT³, then N corresponds to the FFT size or number of points. The number of bands (N/2) can vary from: 4, 8, 16, 32, or 64. For a mono filterbank it is also possible to perform a 128-band filterbank. This parameter (N) sets the frequency resolution (bandwidth) of each band, in relation to the selected sampling frequency. The bandwidth of each band is easily calculated as f_s/N for odd stacking. For even stacking, the bandwidth is also f_s/N except for the DC and the Nyquist bands that have a bandwidth of $f_s/2N$ each (see Figure 1).

Input Block Size (R) and Oversampling (OS)

The input block size is determined by the parameter R, which represents the time-domain resolution. That is, R is the number of new samples included in every transform. When using the WOLA coprocessor, the input block size (R) can be set to 2, 4, 8, 16, 32, 64, or 128. If the WOLA is not used, then the input block size R can be chosen to be 1, which allows performing sample-by-sample time-domain operations. In the WOLA implementation, the input block

size (R) also corresponds to the decimation factor of the filterbank. This means that one sample per band is produced in the frequency domain, when R new input samples enter the transform process.

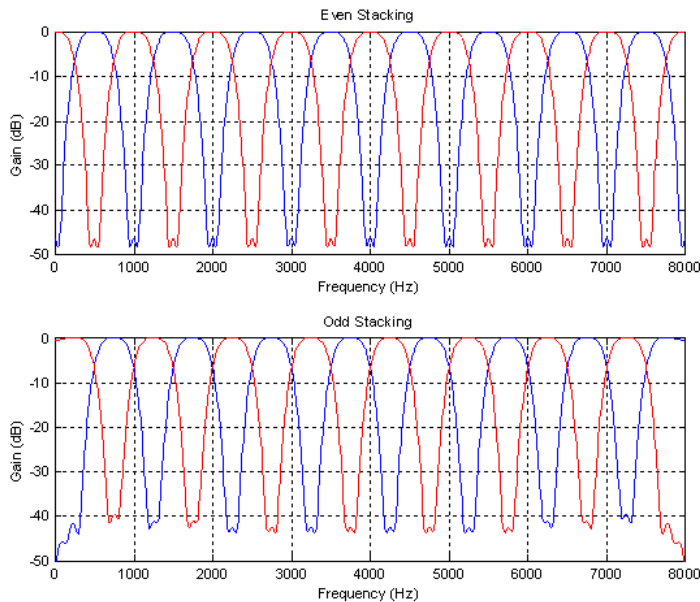
The filterbank oversampling ratio (OS) is defined as N/R . This ratio, OS, plays an important role in controlling the audio quality. For more information, refer to the “WOLA Filterbank Coprocessor: Introductory Concepts And Techniques” tutorial.

Analysis (La) and Synthesis (Ls) Filter Length

The length of the analysis and synthesis filters can be independently configured. Parameter La sets the analysis filter length (32, 64, 128, or 256), while parameter DF (defined as a power of two) sets the length of the synthesis filter, $L_s = L_a / DF$.

Band Stacking (Even/Odd)

The filterbank stacking parameter controls the way the bands are arranged on the frequency axis, as illustrated in . In odd stacking, the $N/2$ bands are uniformly distributed between $f = 0$ and $f = f_s/2$, where f_s is the sampling frequency. In even stacking, the bands are shifted to the left by half the bandwidth of a band. As a result, the first band is centered at the origin (DC) and has half the width of the others. At the other end of the spectrum, another half band appears (the Nyquist band), making the total number of bands to be $N/2 + 1$. Depending on the application, one or the other stacking mode may be advantageous. The odd stacking configuration does require slightly more computation by the WOLA filterbank coprocessor; however, this is not an issue for most applications.



Even Stacking:

$N/2+1$ bands, with a DC and a Nyquist band having half bandwidth.

Odd Stacking:

$N/2+1$ bands, uniformly distributed between $f = 0$ and $f = f_s/2$.

**Figure 1. Frequency Response of Individual Bands for an Even and Odd Filterbank
N = 32 (16 bands) and $f_s = 16$ kHz**

3. This requires a special WOLA microcode, only delivered upon request. Please contact ON Semiconductor support for more information.

CONFIGURATION AND USE OF THE IOP AND WOLA COPROCESSOR

ON Semiconductor provides macros and defines to configure and use the IOP and the WOLA filterbank coprocessor in assembly source code.

```
// Define WOLA parameters
#define WOLA_MODE          WOLA_MODE_MONO          // Mode is MONO
#define WOLA_STACKING     WOLA_STACKING_EVEN      // Stacking is EVEN
#define WOLA_N            64 // Number of band (N/2) is 32
#define WOLA_La          128 // Analysis window length (La) is 128
#define WOLA_OS          8 // Oversampling factor (OS) is 8
#define WOLA_DF          4 // Synthesis window length (La/DF) is 32
#define WOLA_R            (WOLA_N / WOLA_OS )     // Block length is 64/8=8

// Defines the IOP parameters
#define IOP_BLK_NN paste(IOP_BLK_,WOLA_R)
#define IOP_WIN_NN paste(IOP_WIN_,WOLA_La)
#define IOP_CONFIG      ( IOP_ENABLE | \
                          IOP_WIN_NN | \
                          IOP_BLK_NN | \
                          IOP_INPUT_MONO | \
                          IOP_OUTPUT_MONO | \
                          IOP_AUTOMUTE_ENABLE | \
                          IOP_CHAN_INTERLEAVED )
```

The parameter for the input block size, WOLA_R, is not required by the WOLA filterbank coprocessor because it is implicitly defined by the ratio WOLA_N / WOLA_OS. WOLA_R is only used for the IOP configuration. The IOP configuration involves two important parameters:

1. IOP_BLK, the FIFO block size. It must be set to WOLA_R in mono mode and 2 * WOLA_R in stereo mode (2 channels).
2. IOP_WIN, which represents the total number of samples (current block plus previous ones) to be available in the FIFO for the WOLA analysis operation. IOP_WIN must be set to WOLA_La in mono mode, and 2 * WOLA_La in stereo mode.

Configure and Start IOP

The IOP is configured and started by one function. Once started, the IOP will continue to run in the background—no further management is required, unless an audio mute function is desired.

```
Set_IOP_Cfg( IOP_CONFIG ) // Configure and Start IOP
```

Configure and Start the WOLA Coprocessor

Using the WOLA filterbank coprocessor is very simple because the WOLA filterbank is microcode-driven. The following two steps are required in order to use the WOLA filterbank coprocessor:

1. The microcode, the analysis and the synthesis filters (windows) for the selected filterbank must be included with the compiled application and loaded to the appropriate memory location. The WOLA_CONFIGURE () macro, which is provided as part of the firmware, will perform these operations.

Initialization of Parameter Values

Typically, the developer will use #define statements to set the various IOP and WOLA parameters:

```
WOLA_CONFIGURE(WOLA_La, WOLA_N,
               WOLA_OS, WOLA_DF, \
               WOLA_MODE, WOLA_STACKING)
```

This macro will load the appropriate microcode for running the WOLA filterbank coprocessor in the specified configuration, using the supplied default filters (windows). To use custom filters, the dedicated WIN_CONFIGURE_ANALYSIS and WIN_CONFIGURE_SYNTHESIS macros are available. Refer to the Firmware Reference Manual for more information.

During execution of an audio application, the WOLA analysis, gain application and/or synthesis operations can be triggered using the WOLA_Start () macro, by specifying the corresponding parameter. For example:

```
WOLA_Start(UCODE_ANALYSIS_FUNCTION)
WOLA_Start(UCODE_GAIN_FUNCTION)
WOLA_Start(UCODE_SYNTHESIS_FUNCTION)
```

The #defines for UCODE_n_FUNCTION are created by the WOLA_CONFIGURE () macro.

An interrupt is generated at the end of each WOLA operation. When the WOLA interrupt is enabled, the RCore DSP will execute the associated interrupt service routine (ISR). At the end of the WOLA analysis and gain application operations, the results of the WOLA process are available in the dedicated memory locations, and can be used by the RCore. For example, the analysis⁴ results are available at location D_WOLA_RESULT_BASE in X memory. Similarly, the gain application process puts the results back to the same location, D_WOLA_RESULT_BASE, overwriting the analysis results.

4. The format is described in the Hardware Reference Manual.

Other Considerations

WOLA Busy

The status of the WOLA filterbank coprocessor can be determined at any time by reading the WOLA busy flag. Therefore, it is possible to poll this flag to obtain information about the WOLA filterbank coprocessor status.

Note: When the WOLA filterbank coprocessor is busy, reading data from the WOLA shared memory locations (typically `D_WOLA_RESULT_BASE`) is not allowed and will provide erroneous values.

WOLA Interrupt

The WOLA filterbank coprocessor only generates a single interrupt. The interrupt produced by the WOLA filterbank coprocessor is the same when the analysis, gain application or synthesis operations are completed. Consequently, the same Interrupt Service Routine (ISR) is run after each operation. This means that code in the ISR must track the WOLA state. Usually, a memory location containing the current WOLA state number (0 for analysis, 1 for gain application, 2 for synthesis, and sometimes 3 for idle) is used for this purpose.

Efficiency in Data Access

For efficiency, the WOLA filterbank coprocessor works in a closed manner with the IOP. When starting an analysis, the input data (R samples) are always taken from the input FIFO, at location `D_SMART_IN_FIFO_BASE` in X memory. Similarly, after synthesis, the output R samples coming out of the WOLA filterbank coprocessor are directly copied to the output FIFO, at location `D_SMART_OUT_FIFO_BASE`, in the Y memory. Whenever a time-domain pre-processing function is required, the programmer must place the pre-processed data back into the input FIFO before running the WOLA analysis. Similarly, any post-processing performed after WOLA synthesis should place the post-processed data back into the output FIFO.

Block Floating-Point

The WOLA filterbank coprocessor uses block floating-point arithmetic to maintain high numeric precision in the audio signal path, by preventing saturation (or overflow) of the data during internal filterbank calculations. After analysis, the scale of the analysis results is dependent upon the input signal level, frequency composition, phase, and number of bands. This scale factor may change from one processed block of data to the next. Consequently, when performing calculations based upon the analysis results, it is necessary to account for this scaling.

The `D_BLOCK_EXP_DATA` register always contains the number of left shifts to apply globally to all bands, in order to restore the correct level. This compensation has to be done in all situations requiring true scale spectra after analysis, or coherent scales throughout successive transforms. No scaling is required after synthesis.

Gain Application

Two steps occur during the WOLA gain application:

1. The analysis results are multiplied by gain values associated to every band, which are stored at location `D_GAIN_BASE` in X memory, and
2. A left shift by the value stored in `D_GAIN_EXP_DATA` is applied to globally modify the results.

The gain multiplier values are represented as fractional numbers and represent values in the range $[-1, +1]$. Gains of greater than unity can be achieved by applying the number of left shifts stored in the `D_GAIN_EXP_DATA` register. The `D_GAIN_EXP_DATA` value can vary from 0 to 15 and in theory 90 dB of gain can be applied. However, depending on the level of the input signal, saturation will occur if more gain is applied than the signal (analysis results values) can accommodate. The WOLA filterbank has 2-bits of dynamic range headroom. When the modified analysis data reach their maximum range (i.e., 16 bits), problems can occur if the value in register `D_GAIN_EXP_DATA` is higher than 2 (representing a gain higher than 4). Data values that would exceed the maximum 18-bit range are saturated to the maximum full-scale value during WOLA gain application. The saturation occurs independently for each band and each real and complex value. Furthermore, after synthesis, the audio output sample data is limited to a maximum dynamic range of 16 bits⁵.

Certified Microcode

ON Semiconductor provides microcode for a large number of WOLA filterbank configurations. Not all of the supplied configurations will be suitable for an intended audio application. To help the user in selecting a particular configuration, ON Semiconductor publishes a list of certified microcode that will meet certain target audio performance criteria. The criteria for certification are:

- 30 dB spurious free dynamic range (SFDR)
- Less than 1 dB pass band ripple
- All WOLA operations must be completed within a time frame that is less than 80% of the RCore cycles available per block for a 1.28 MHz clock

5. Refer to the application note "Using Block Floating Point in the WOLA Filterbank Coprocessor" for more information about the block floating-point mechanism

CONSIDERATIONS FOR BLOCK-BASED PROCESSING

Frame (Block) Rate

The IOP generates an interrupt (called IOP_Block_Full) every time the number of samples corresponding to the input block size is accumulated. The frame rate is determined by the amount of time from one IOP_Block_Full interrupt to the next IOP_Block_Full interrupt. In a typical implementation, for each block of audio data, the WOLA filterbank coprocessor performs three operations (analysis, gain application and synthesis),

which must be completed within one frame. The WOLA filterbank coprocessor will generate an interrupt each time an operation is completed. Simultaneously, the RCore DSP is performing operations and managing interrupts. Figure 2 illustrates the typical program structure, showing the three processors (IOP, RCore and WOLA filterbank), tasks, and associated interrupts. The activity for one block of audio data (also called tick) is shown. In this example, a tick is defined as the time duration between the start and completion of the arrival of an input block (R new audio samples).

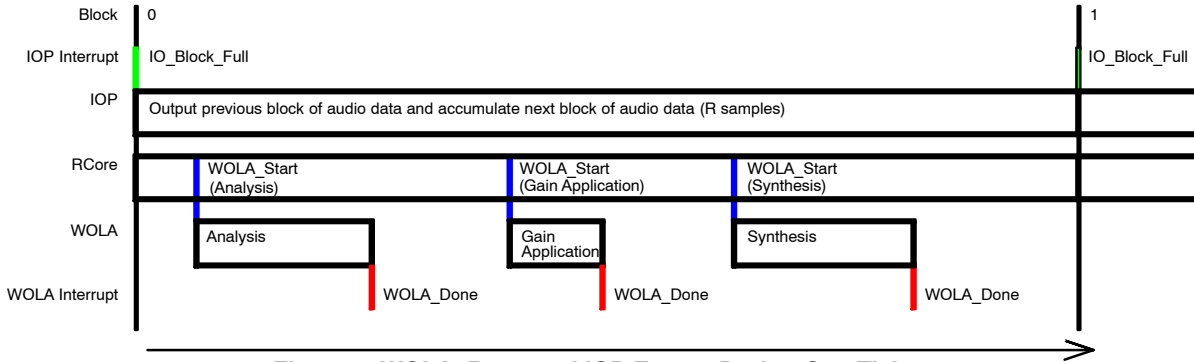


Figure 2. WOLA, RCore and IOP Events During One Tick

The duration of one tick depends on the input block size R and on the sampling frequency, according to the following expression:

$$\text{Tick [ms]} = \frac{R[\text{samples}]}{fs [\text{kHz}]}$$

For example, the tick duration for an input block size of 8 samples and a 16 kHz sampling frequency, is 0.5 ms.

The following steps are typically followed during one tick:

1. **Beginning of the Tick:** The IOP sends an interrupt (IO_Block_Full) to the RCore, indicating the presence of new data. At that time, the new samples are available in the input FIFO (at location D_SMART_IN_FIFO_BASE), and time-domain pre-processing could be performed if required, before starting WOLA analysis. Processed samples would have to be put back into the FIFO at the same location.
2. **WOLA Analysis:** The RCore launches the WOLA filterbank coprocessor analysis operation. While the WOLA filterbank coprocessor is working, the RCore can proceed with other tasks in parallel. Once the analysis is completed, an interrupt is generated by the WOLA filterbank coprocessor (WOLA_Done). The results of the analysis are then available in memory (at location D_WOLA_RESULT_BASE) for processing by the RCore (typically for determining the new gains to be applied later by the WOLA filterbank coprocessor). The value of

D_BLOCK_EXP_DATA must also be used to correctly scale the analysis results, if required.

3. **Gain Application:** The RCore then launches the next WOLA operation, the gain application. During the WOLA gain application, the analysis results are multiplied by the WOLA gains (as previously determined by the RCore). The gains are stored from location D_GAIN_BASE before running the gain application, while the analysis results are taken from their reserved location (D_WOLA_RESULT_BASE). The resulting values are then globally scaled by the content of the register D_GAIN_EXP_DATA. While the WOLA filterbank coprocessor is working, the RCore can proceed with other tasks to be run in parallel. As soon as the gain application is completed, an interrupt is generated by the WOLA (WOLA_Done). The gain-modified analysis results are then available at location D_WOLA_RESULT_BASE, replacing the previous analysis results.
4. **Synthesis Operation:** The RCore then launches the WOLA filterbank coprocessor synthesis operation, which takes the frequency-domain data at D_WOLA_RESULT_BASE and re-transforms them into time-domain output samples. As usual, while the WOLA filterbank coprocessor is working, the RCore can proceed with other tasks to be run in parallel (typically gain calculation for

the next tick). The last interrupt from the WOLA filterbank coprocessor is then received (WOLA_Done), indicating that synthesis is complete.

Time-domain samples are then available in the output FIFO (at location D_OUT_SMART_FIFO_BASE). Any additional time-domain processing can be performed as required, just before the end of the tick. Processed samples would have to be put back into the output FIFO.

5. **End of the Tick:** The end of the tick is finally reached when the new IOP interrupt occurs. New input data is available in the input FIFO (D_SMART_IN_FIFO_BASE) for processing at the next tick and the samples previously stored in the output FIFO are sent to the audio output stage.

Cycles in a Tick

Together with the system clock (SYS_CLK), the tick duration is a major parameter of the application because it sets the number of processing cycles available to perform the computations for one block of R audio samples. The number of processing cycles available to the R_{Core} during one tick can be calculated by the following expression:

$$\text{Nbr. Cycles Per Tick [cycles]} = \text{Tick [ms]} \times \text{SYS_CLK [kHz]}$$

For example, 640 processing cycles are available for processing one block of data in the R_{Core}, when the tick duration is 0.5 ms, at a 1.28 MHz system clock speed. Naturally, doubling the system clock to 2.56 MHz would provide 1280 cycles, allowing more processing to be performed on one block of data.

The total number of cycles required for the WOLA filterbank to perform analysis, gain application and synthesis operations should not be greater than the number of cycles available to the coprocessor in one tick. In fact, extra cycles may be required if the R_{Core} is to apply any pre- or post-processing or handle any possible interrupts. The number of cycles required by the WOLA filterbank coprocessor for typical configurations is available in an Excel spreadsheet provided with this application note.

Using Toccata Plus and BelaSigna 200, increasing the system clock implies that more cycles will be available to the WOLA filterbank coprocessor because the WOLA filterbank is driven by the same clock (SYS_CLK). As a consequence, the WOLA filterbank coprocessor will take less time (but the same number of cycles) than before to perform its transform tasks. Increasing the WOLA clock (WOLA_CLK) does not benefit the coprocessor as it has already enough time to execute the complete set of operations. However, more cycles become available to the R_{Core} to apply time-domain pre- or post-processing in the tick.

The Orela 4500 series offers the possibility to set the WOLA_CLK independently of the R_{Core} clock (SYS_CLK). In this way, the time duration during which the WOLA filterbank coprocessor is active can be tuned independently of the R_{Core} clock (SYS_CLK).

Group Delay

The IOP/WOLA filterbank coprocessor group delay is the amount of time it takes the audio signal to pass from the input stage through the WOLA filterbank coprocessor to the output stage. The IOP and the WOLA filterbank coprocessor have the following sources of delay:

- The delay generated in the WOLA analysis filter.
- The delay generated in the WOLA synthesis filter.
- The block pipelining of the audio data by the IOP.

The analysis and synthesis filters are finite impulse response (FIR) structures. In terms of number of samples, they introduce an algorithmic delay equal to the sum of half the filter lengths. Thus, the analysis filter introduces a delay equal to $L_a/2$ samples, while the synthesis filter additionally introduces a delay equal to $L_s/2$. Finally, since analysis and synthesis overlap within one tick, there is a one block reduction in the delay (R). The IOP introduces a two block pipelining delay ($2 * R$ samples). Therefore, the total group delay through the WOLA filterbank coprocessor is:

$$\text{Group Delay} = \frac{L_a}{2} + \frac{L_s}{2} + R \text{ samples}$$

Assuming a sampling frequency f_s expressed in kHz, this delay corresponds to $(L_a/2 + L_s/2 + R)/f_s$ milliseconds.

WOLA FILTERBANK PARAMETER SELECTION

The selection of appropriate WOLA filterbank parameters requires consideration and compromise between desired frequency and time resolutions, acceptable group delay, audio performance, computation power, and of course application features. During the design of the WOLA filterbank configuration, one usually needs to check the performance of the filterbank. The key WOLA performance characteristics are audio performance, group delay and power consumption. The audio performance is usually evaluated, on one hand by the level of ripples produced in the passband, and on the other hand, by the level of aliasing and image components present after reconstruction. The level of ripples can be assessed by processing an impulse through the WOLA analysis and synthesis chain, and calculating the frequency response. The level of aliasing and imaging components or Spurious-Free Dynamic Range (SFDR) can be estimated by keeping only one band during synthesis (setting all gains in other bands to 0). In this way the alias and image components in the other areas of the spectrum can be observed.

This section provides some general guidelines, showing the respective influence of various parameters on performance characteristics⁶.

6. No discussion of filter selection is provided in this document. It is assumed that the default filters are used. Information about filter design can be found in the WOLA Filterbank Coprocessor: Introductory Concepts And Techniques tutorial.

Examples

Consider the following 8-band configuration, with a sampling frequency of 16 kHz:

- R = 2
- N = 16
- La = 32
- DF = 2
- Stacking: odd
- As a result:
- OS = N/R = 8
- Ls = La/DF = 16

This configuration has very good audio properties (because the oversampling ratio is high), and it provides a very low delay (short filters and block size):

$$\text{Delay} = \left(\frac{La/2 + Ls/2 + R}{16} \right) = \frac{(16 + 8 + 2)}{16} = 1.625 \text{ ms}$$

The frequency response is illustrated in the first plot of , when only the gain in the sixth band is equal to one, the others being set to zero. The first plot shows the case where all the gains are set to 0, except for the sixth band. The second plot shows the situation where the gains are set to 1 in all bands. In both cases, the vertical dotted lines represent the band limits.

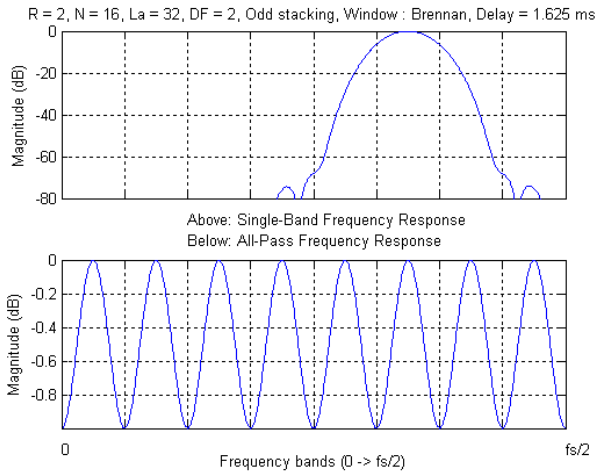


Figure 3. Frequency Response of the WOLA Filterbank Process in Configuration Considered (Analysis + Gain Application + Synthesis)

Impact on Audio Performance

Better audio performance (increased SFDR) can be obtained by:

- Increasing the analysis filter length (La).
- Increasing the synthesis filter length (Ls), that is, decreasing DF.
- Reducing the block size (R), or increasing the oversampling ratio (OS).

Impact on Group Delay

The group delay $(La/2 + Ls/2 + R)/fs$ can be decreased by:

- Reducing the analysis filter length (La).
- Reducing the synthesis filter length (Ls), that is, increasing DF.
- Reducing the block size (R).
- Increasing the sampling frequency (fs).

Impact on Power Consumption

Power consumption can be reduced by:

- Reducing the analysis filter length (La).
- Reducing the synthesis filter length (Ls), that is, increasing DF.
- Increasing the block size (R).
- Decreasing the oversampling ratio (OS).
- Decreasing the sampling frequency (fs).

Guidelines

1. To maintain similar audio performance, while increasing the number of bands (N/2), increase the filter lengths (La, Ls). Alternatively, the oversampling factor (OS) could be held constant while increasing R.
2. To decrease the group delay with minimal impact on the audio quality, decrease the length of the synthesis filter (Ls) as compared to the length of the analysis filter (La) i.e., increase DF.
3. The decimation factor (DF) should not be higher than the oversampling ratio (OS). In most cases, if DF is greater than OS, then the synthesis filter will not correctly remove the imaging distortion introduced during synthesis.
4. Configurations with $N = La$ (usually associated with $La = Ls$ and $DF = 1$) should generally be avoided. They are not appropriate when large gain adjustments have to be applied.
5. Setting OS to 4 or more is generally the best way to maintain good audio quality.

This filterbank configuration has a very smooth and wide transition from one band to the next band. While no aliasing is produced thanks to the large oversampling ratio (the images are far away from each other on the frequency axis), applying extremely sharp gain characteristics in this configuration would have reduced effects, because of the very large transition bands. For example, setting the gains 1, 0 and 1 in bands 3, 4 and 5 (respectively), would not appropriately remove the components in band 4, because those components would be provided by bands 3 and 5. This situation could be improved by using a larger filter ($La = 64$, $Ls = 32$), increasing the computation (power) and compromising the group delay, which would become $(32 + 16 + 2)/16 = 3.125$ ms.

The second plot in shows that the range of the pass band ripples is less than one decibel.

The second example shows a configuration with 16 bands instead of 8. Applying the guidelines, it appears that the filter lengths should be increased. Typically, we could select:

$$R = 2$$

$$N = 32$$

$$L_a = 64$$

$$DF = 2$$

Stacking: odd

As a result:

$$OS = N/R = 16$$

$$L_s = L_a/DF = 32$$

The oversampling factor is very high (16) and it may be reasonable to increase R in order to decrease computational load (and power consumption). Using $R = 4$ instead of 2 would be a satisfying solution (delay 3.25 ms), providing similar aliasing properties to the previous example. However, the filter transition behavior is relatively poor for the number of bands. The filter transition could be improved by increasing the analysis filter length to 128, but keeping L_s to 32 (changing DF to 4 instead of 2) in order to keep a low delay: $(64 + 16 + 4)/16 = 5.25$ ms. In Figure 4, the first plot shows the case where all the gains are set to 0, except for the sixth band. The second plot shows the situation where the gains are set to 1 in all bands. In both cases, the vertical dotted lines represent the band limits.

$$R = 4$$

$$N = 32$$

$$L_a = 128$$

$$DF = 4$$

Stacking: odd

$$OS = N/R = 8$$

$$L_s = L_a/DF = 32$$

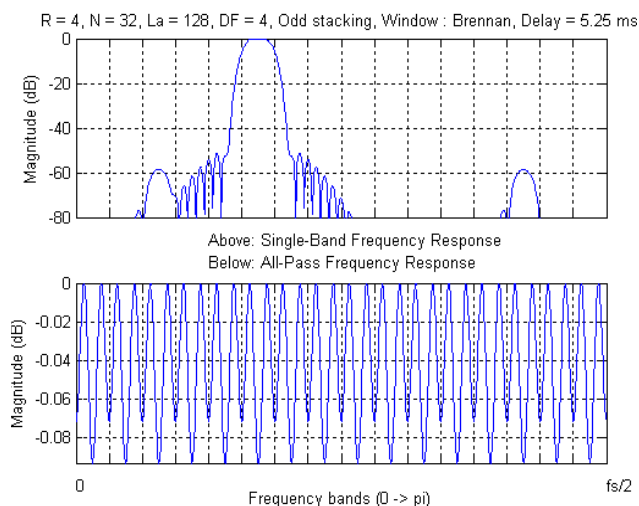


Figure 4. Frequency Response of the WOIA Filterbank Process in the Configuration Considered (Analysis + Gain Application + Synthesis)

Finally, to increase the number of processing cycles available per tick, this configuration could be modified, by increasing the block size to 8. The result is a minor reduction in audio quality or SFDR (still acceptable because the oversampling ratio is 4), and increased group delay (5.5 ms). By increasing the block size more calculations are possible within a tick, while reducing the time resolution.

The figures in this section were generated using the WOLA toolbox for MATLAB, which is part of ON Semiconductor's simulation tools.


SUMMARY AND FURTHER READING

This application note introduces the IOP and the WOLA filterbank coprocessor, describing their features and typical usage. It also provides an introduction to considerations in WOLA filterbank parameter selection and compromises for specific applications.

The following additional information and/or tools are also available:

- **WOLA Filterbank Coprocessor: Introductory Concepts and Techniques Tutorial:** This tutorial describes the signal processing aspects of the WOLA filterbank coprocessor, in order for the designer to fully understand the WOLA mechanisms. It includes many examples.
- **Using Block Floating-Point in the WOLA Coprocessor:** This application note describes how to correctly deal with block floating point in order to avoid saturation and overflow problems.
- **WOLA Toolbox for MATLAB:** This software provides WOLA C-callable and MATLAB functions, in order for the user to include the particular WOLA processing in their algorithm simulation.
- **SignaKlara® Blockset for Simulink:** This is a simulation tool that uses Simulink and WOLA toolbox for MATLAB.
- **Hardware Reference Manual:** Chapter 5 in this manual describes how to configure and use the WOLA filterbank coprocessor in detail.
- **WOLA Excel Spreadsheet:** This provides the number of processing cycles for every WOLA configuration.
- **WOLA Certified Configurations:** This lists the most usual WOLA configurations (the ones included in ON Semiconductor's EDKs), which provide a certain level of audio quality.

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 and Simulink are registered trademarks 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
Order Literature: <http://www.onsemi.com/orderlit>
For additional information, please contact your local
Sales Representative