Seamlessly Interfacing MEMS Microphones with Blackfin® Processors

Contributed by Nagaraj Hegde

Rev 1 – August 3, 2010

1 Introduction

This EE-Note describes a novel way of interfacing ADI's industry-leading high-performance ADMP421 MEMS microphone to the Blackfin® processor family. The application demonstrates a software implementation of Cascaded Integrated Comb (CIC) decimation filters. This eliminates the necessity of any additional audio ADC components as Blackfin processors can directly take the Pulse Density Modulated (PDM) data coming out of the microphone and convert it to I2S format. Blackfin processor's DSP capabilities, along with readily available software modules for a variety of audio algorithms, reduce the hardware and software overhead in system designs that use microphones. Using Blackfin processors in the system, designers can achieve low power and high performance.

MEMS microphones in the digital audio segment of the industry are swiftly making an entry into applications such as cell phones, PCs, digital cameras, and Bluetooth® headsets. The Blackfin family is Analog Devices high performance processors, widely used in embedded signal processing applications.

2 ADMP421 MEMS Microphone

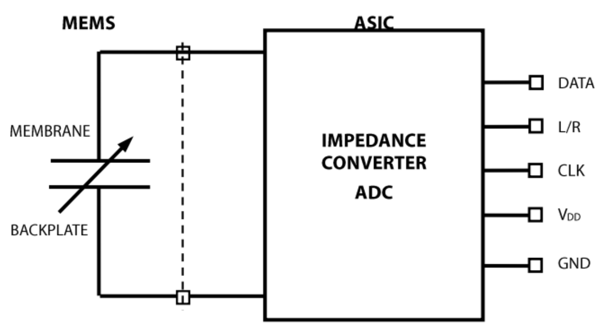

Figure 1. ADMP421 functional block diagram

The ADMP421 device is a low-cost, low-power, digital output bottom-ported omni-directional MEMS microphone. The ADMP421 part consists of a MEMS microphone element, an output amplifier and a 4th order sigma delta modulator. The digital interface allows for the PDM (Pulse Density Modulated) output of two microphones to be time multiplexed on a single data line using a single clock. The method for this will be explained further.

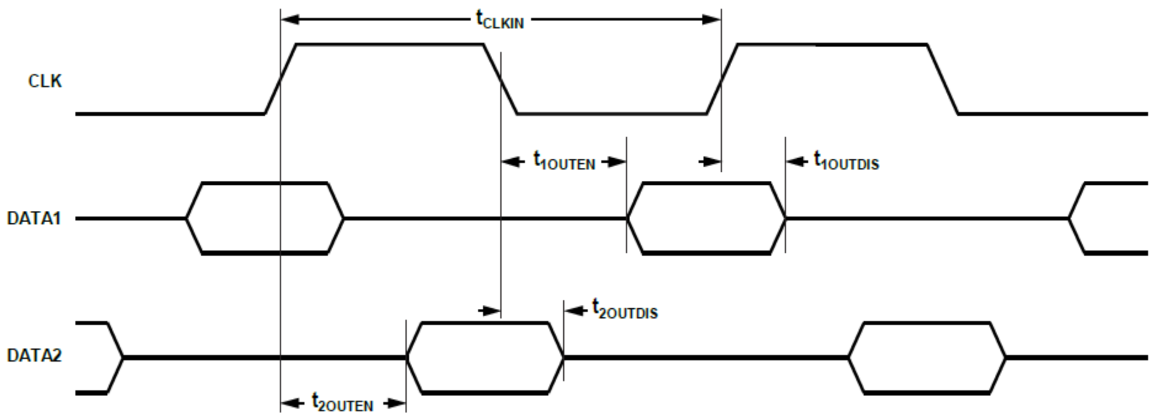

Figure 2. Timing diagram of the data driven by the microphone

Figure 2 shows the timing of the data driven by the microphones with respect to the clock based on the `L/R` pin of the microphone. If the `L/R` pin is tied to the `GND` pin, the data is driven on the rising edge of the clock. If the `L/R` pin is tied to `VDD` pin the data is driven on the falling edge of the clock.

The ADMP421 device has a high signal-to-noise-ratio (SNR) and high sensitivity, making it an excellent choice for far field applications. The ADMP421 part has a flat wideband frequency response, resulting in natural sound with high intelligibility. Low current consumption and a sleep mode enable long battery life for portable applications. A built-in particle filter provides for high reliability. The ADMP421 device complies with the *TIA-920 Telecommunications Telephone Terminal Equipment Transmission Requirements for Wideband Digital Wire Line Telephones* standard.

3 Introduction to Pulse Density Modulation[2]

Pulse density modulation, or PDM, is a form of modulation used to represent an analog signal in the digital domain. In a PDM signal, specific amplitude values are not encoded into pulses as they would be in PCM. Instead it is the relative density of the pulses that corresponds to the analog signal's amplitude.

In a pulse density modulation bit stream a "`1`" corresponds to a pulse of positive polarity (`+A`) and a "`0`" corresponds to a pulse of negative polarity (`-A`). Mathematically, this can be represented as:

$$x[n] = -A(-1)^{a[n]}$$

Equation 1. Pulse density modulation

Where `x[n]` is the bipolar bit stream (either `-A` or `+A`) and `a[n]` is the corresponding binary bit stream (either `0` or `1`).

To get the framed data from the PDM bit stream, decimation filters are usually used in sigma delta analog-to-digital converters (ADCs). A widely adopted approach in this context is using CIC filters (also known as SINC filters) at the first stage of decimation to reduce the sampling frequency, followed by 2:1 half-band low-pass decimation filters to take out the high frequency noise introduced in the sigma delta modulation process and the further decimation. This EE-Note demonstrates a way to achieve a seamless interface between a Blackfin processor and the MEMS microphones by implementing the above-mentioned filter functionalities in software.

4 System Level Design

A system level block diagram of the set up is shown in Figure 3 below. The data coming out of the microphone is sent to the decimation process, which consists of three parts: a CIC decimation filter converting 1-bit PDM data to framed data, followed by two 2:1 half band filters and an FIR filter in the final stage eliminating the high frequency noise generated in the sigma delta modulation process in the Microphone. The reconstructed audio is sent to a DAC for audio output purpose.

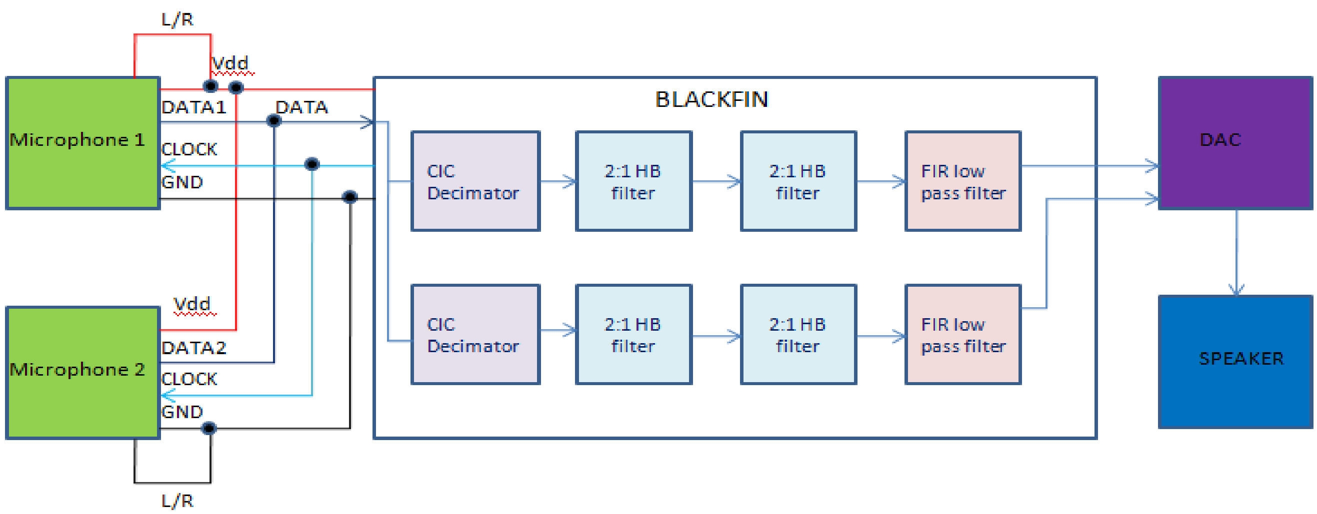

Figure 3. Interfacing two MEMS microphones to a Blackfin processor

4.1 Hardware Interface

The ADMP421 MEMS microphone is interfaced to the Blackfin processor over the Serial Port (SPORT). The microphones can drive the PDM data on either rising edge or falling edge of the clock based on the logic level at the L/R select pin. Interfacing a single microphone to the DSP is straightforward. All that needs to be done is to provide the same clock (in the range of 1 to 3 MHz) to the microphone and the SPORT and receive the PDM data into the processor from the SPORT, while keeping the L/R pin tied to GND or VDD. To connect two such microphones to a single serial port data line, the L/R select pin of one microphone has to be grounded. The L/R select pin of the other microphone must be connected to VDD. This ensures that the microphones drive data on opposite edges of clock. To make the SPORT receive data from both microphones, the microphones have to be clocked at half the rate of the clock at which the SPORT Rx is running. Either an external clock source or the SPORT itself can generate these clocks. The microphone modulates audio signals with respect to the clock fed to it.

4.2 Software Routines

The frequency of the PDM data output from the microphone (which is the clock input to the microphone) must be a multiple of the final audio output needed from the system. For example, in the current implementation, we are doing a decimation of 32; for the output rate of 96 kHz, we need to provide a clock frequency 3.072 MHz to the microphone. For a two-microphone interface, the clock to the microphones remains the same but the SPORT should run at double the frequency of the microphones to catch bits from both microphones. In either case, the software routines should operate on bits (not words) as the data coming out of MEMS microphone is 1 bit PDM stream. For a two-microphone interface, the data in the receive buffer will interlaced bit by bit. Therefore, software routines must ensure that the data from two microphones are processed separately. Since the sigma-delta modulator inside the microphone is of the 4th order, a 5th order CIC decimator is implemented in the Blackfin processor to reconstruct the audio bit stream. The CIC stage will be followed by two 2:1 half-band decimation stages and then low-pass filtered to get framed audio data. System designers can add custom audio algorithms (such as audio compression) on the audio data taken from microphones, after the FIR stage. The example code included in the associated `.ZIP` file provides details on the described implementation methods.

4.3 Identifying the Source Microphones of the Interlaced Data

For a two-microphone interface, the data inside the receive buffer will be interlaced bit by bit. This means that every alternate bit belongs to the same microphone. To understand which microphone the alternate bits belong, you need only to connect the microphone's clock (which is one half of the clock at which the SPORT is running) to the SPORT frame sync pin and run the SPORT in unframed mode. The SPORT only starts receiving the data from the first rising edge of the microphone's clock, and this will be the data from that microphone, whose `L/R` select pin is shorted to `VDD` (which means that the microphone drives the data on the rising edge of the clock). For details, refer to the ADMP421 device data sheet [5]). So, in all cases, the data stream inside the data buffers starts from the data from this microphone, and the software routines are written such that data from different microphones are processed separately.

4.4 CIC Filters

Cascaded Integrator-Comb (CIC) filters are a class of linear phase FIR filters comprised of a comb part and an integrator part. The transfer function of a CIC decimator filter is [1]:

$$H(z) = H_I^N(z)H_c^N(z) = \frac{\left(1-z^{-RM}\right)^N}{\left(1-z^{-1}\right)^N} = \left[\sum_{k=0}^{RM-1} z^{-k}\right]^N$$

Equation 2. CIC decimator filter transfer function

Where:

H_I is the transfer function of the integrator part of the filter.

H_C is the transfer function of the comb part of the filter.

N is the number of sections. The number of sections in a CIC filter is defined as the number of sections in either the comb part *or* the integrator part of the filter, not as the total number of sections throughout the entire filter.

R is the decimation factor.

M is the differential delay.

Figure 4 shows the block diagram of a 5^{th} order CIC decimation filter used for reconstructing the audio, which comprises integrators, comb filters, and decimator.

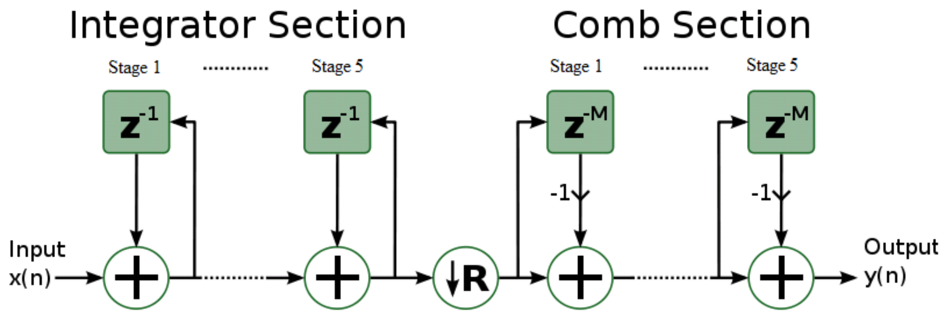

Figure 4. 5^{th} order CIC decimator block diagram

Figure 5 and Figure 6 taken using MATLAB® software show the magnitude and frequency responses of the CIC decimation filter with decimation factor 8 and order 5.

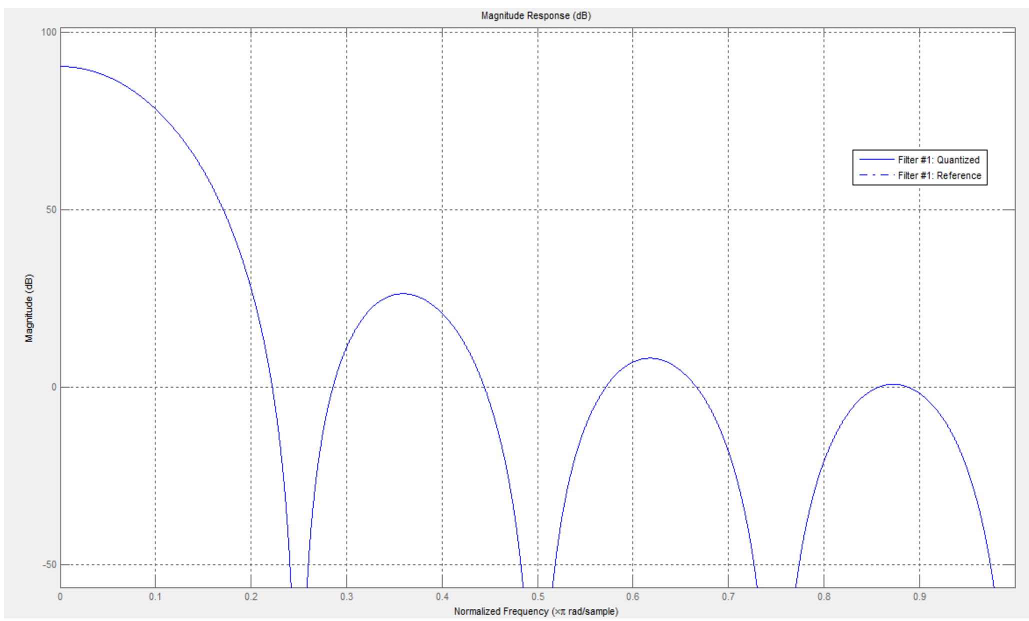

Figure 5.Magnitude response of CIC filter

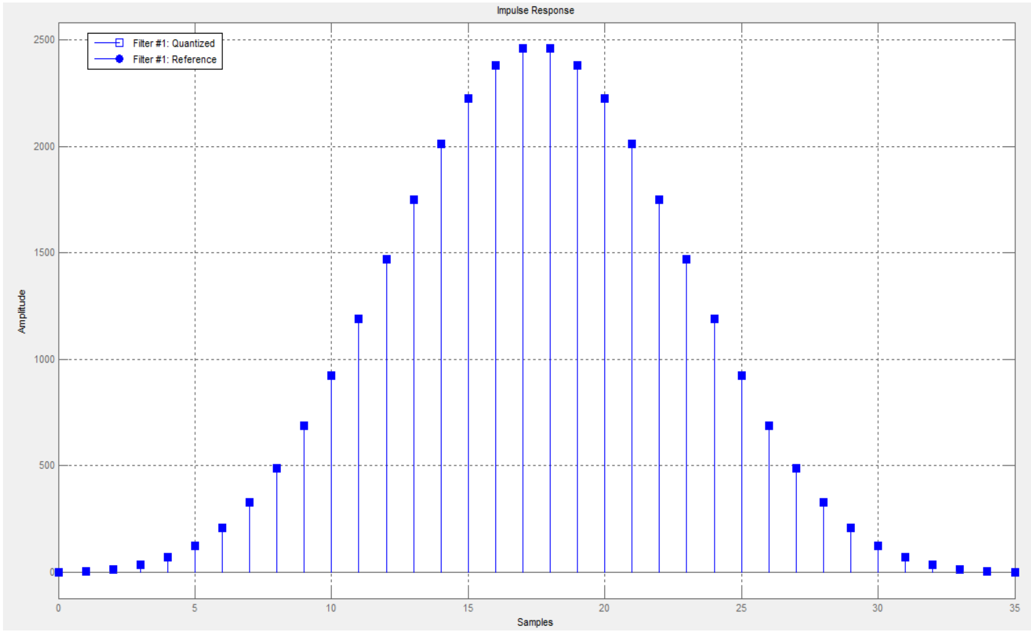

Figure 6. Frequency response of CIC filter

4.4.1 Software Implementation of CIC Decimation Filter

The CIC decimation filter, which plays the major role in the application, has been implemented in C language. The snippet below explains the code flow:

```
/* Below is The CIC decimation filter functionality which gets called after every
time one input buffer is ready for the processing*/

/* loop over 16 bits and operate on every alternate bit for two microphones which
means an operation of divide by 8 decimation */

/* GetBit() functions extracts each bit from the received PDM data in the buffer */

//Sigma operations for the PDM data from L Microphone

    for (i =0; i<16; i=i+2)

             {
                Sigma_L_1 += GetBit(buff,j,i);
                Sigma_L_2 += Sigma_L_1;
                Sigma_L_3 += Sigma_L_2;
                Sigma_L_4 += Sigma_L_3;
                Sigma_L_5 += Sigma_L_4;
             }
//Delta operations for the PDM data from L Microphone
                Delta_L_1 = Sigma_L_5 - OldSigma_L_5;
                Delta_L_2 = Delta_L_1 - OldDelta_L_1;
                Delta_L_3 = Delta_L_2 - OldDelta_L_2;
                Delta_L_4 = Delta_L_3 - OldDelta_L_3;
                Result1   = Delta_L_4 - OldDelta_L_4;
```

```
/* framed data from L microphone to be sent for FIR decimation is stored in Result.
Since the operation is 5 stage divided by 8 decimation this field can grow upto 13
bits. In case the designers want higher decimation than 8 bits (say 16) in the CIC
stage, they need to take care of the register growth of the CIC registers. */

    input1[j] = Result1;

/*Same procedure has to be applied on the data from the other microphone as well;
Refer the code attached with the application note for complete filter function */

    }
```

Listing 1. CIC decimation filter code snippet

4.5 Half-Band Filtering

Half-band filters are widely used in multi-rate signal processing applications when interpolating/decimating by a factor of two.

Half-band filters have two important characteristics: the passband and stopband ripples must be the same, and the passband-edge and stopband-edge frequencies are equidistant from the half-band frequency pi/2.

4.6 FIR Low-Pass Filter

Low-pass filtering must occur after the CIC stage to remove the high frequency noise introduced by the analog-to-digital conversion process in the microphone. MATLAB was used to generate the FIR filter coefficients and the optimized library functions for a 16-bit FIR operating on fractional values in Blackfin processors used for the filtering purpose. Figure 7 shows the frequency response for the 6-kHz FIR low-pass filter with 297 taps.

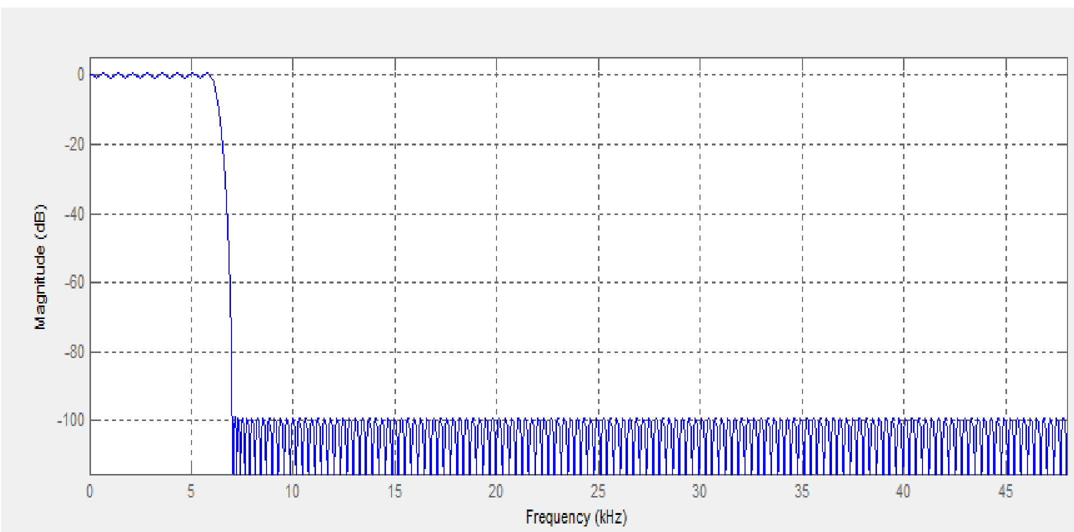

Figure 7. Magnitude response of FIR low-pass filter

5 Experimental Results

The experimental set up consists of two MEMS microphones. Both microphones are connected to the Blackfin SPORT interface. The ADSP-BF533 EZ-KIT Lite® evaluation platform has been selected for interfacing to the microphones. Having said this, any of the existing Blackfin evaluation boards can be used for this same purpose.

The microphones are clocked at 3.072 MHz, while the SPORT receives the data at 6.144 MHz. The data stream is operated on by a divided-by-8 CIC decimator and the received PDM data is being sent out in 13-bit framed data at 384 kHz. This is followed by two 2:1 FIR half-band decimators. The final stage is a low-pass FIR at 96 kHz.

To have the complete real-time implementation of the algorithm, the availability of descriptor-based DMA option in the Blackfin processor proved to be handy. For more information on the implementation, refer to the code associated with this EE-Note.

With the Blackfin core running at 594 MHz, Table 1 provides the number of clock cycles required to execute the various filters per sample, corresponding to a one-microphone interface and a two-microphone interface:

Filters	Number of Core Clock Cycles for One Microphone Interface	Number of Core Clock Cycles for Two-Microphone Interfaces
Divided-by-8 CIC decimator	560	1320
25 tap half-band 2:1 decimator	20	20
FIR low-pass filters with 300 taps	36	36

Table 1. Blackfin core clock cycles to perform filter operations

The figures below show the PDM data taken from the microphone after feeding it with the sine tone and the waveforms at every stage of the decimation process. The figures are taken from a VisualDSP++® plot window.

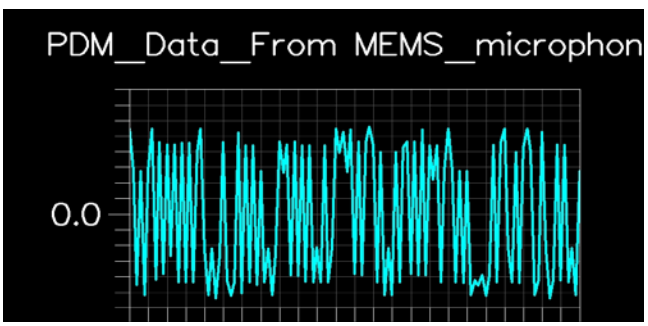

Figure 8. PDM data coming out of microphone

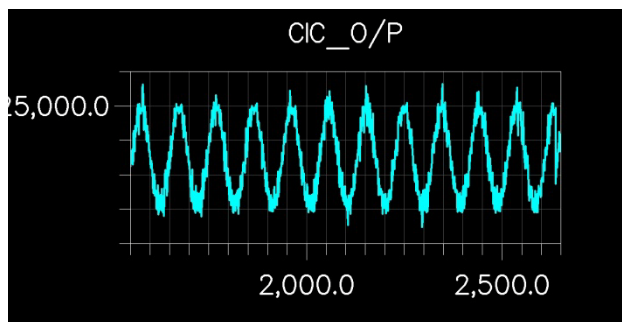

Figure 9. Output of the CIC decimation stage

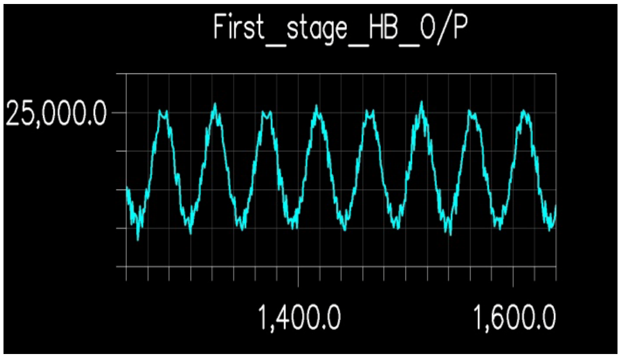

Figure 10. Output of the first stage half-band decimation

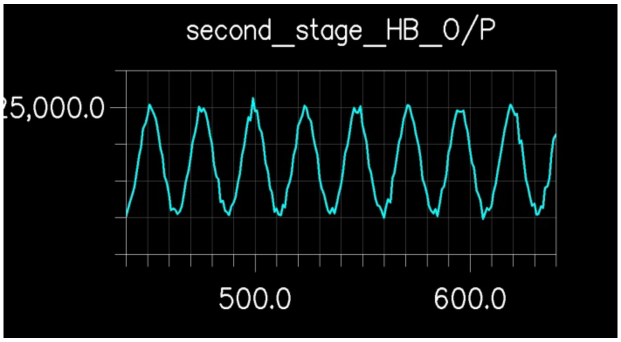

Figure 11. Output of the second stage half-band decimation

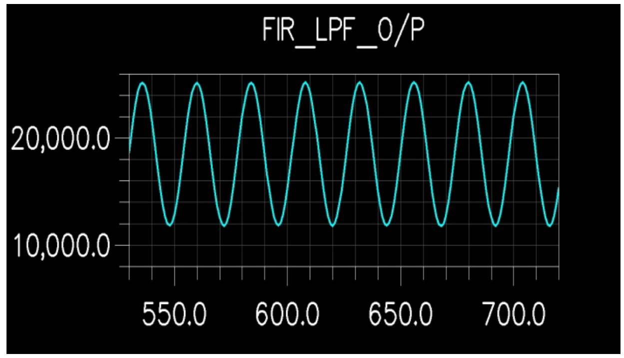

Figure 12. Output of the final FIR low-pass stage

This document demonstrates a seamless interface of a MEMS microphone with the Blackfin processor. Since the hardware interface is over the DSP serial port, the same concept can be extended to interface a MEMS microphone to the Analog Devices SHARC® family of processors as well. The newer ADSP-214xx SHARC family may be attractive in terms of cycle counts due to the processor's hardware accelerators, which can be utilized to perform the low-pass FIR filtering in parallel, while the other functions are being computed.

References

[1] *Eugene B. Hogenaeur, "An Economical Class of Digital Filters for Decimation and Interpolation" IEEE transactions on acoustics, speech, and signal processing, vol. assp-29, no. 2, April 1981.*

[2] *http://en.wikipedia.org/wiki/Pulse-density_modulation*

[3] *ADSP-BF531/ADSP-BF532/ADSP-BF533: Blackfin Embedded Processor Data Sheet.* Rev G, May 2010. Analog Devices Inc.

[4] *ADSP-BF533Blackfin Processor Hardware Reference.* Rev 3.4, April 2009. Analog Devices Inc.

[5] *ADMP421 Omnidirectional Microphone with Bottom Port and Digital Output Data Sheet.* Rev 0, April 2010. Analog Devices Inc.

Document History

Revision	**Description**
Rev 1 – August 3, 2010 *by Nagaraj Hegde*	Initial release.