



Audio Engineering Society Convention Paper

Presented at the 120th Convention
2006 May 20–23 Paris, France

This convention paper has been reproduced from the author's advance manuscript, without editing, corrections, or consideration by the Review Board. The AES takes no responsibility for the contents. Additional papers may be obtained by sending request and remittance to Audio Engineering Society, 60 East 42nd Street, New York, New York 10165-2520, USA; also see www.aes.org. All rights reserved. Reproduction of this paper, or any portion thereof, is not permitted without direct permission from the Journal of the Audio Engineering Society.

High Performance Real-Time Software Asynchronous Sample Rate Converter Kernel

Thierry HEEB¹

¹ ANAGRAM Technologies SA, ZI Le Trési 6A, CH-1028 Préverenges, Switzerland
theeb@anagramtech.com

ABSTRACT

A scalable real-time asynchronous sample rate converter software kernel is presented that offers a flexible alternative to the usual hardware implementations. The kernel is dynamically configurable at run-time and supports almost arbitrary up-sampling or down-sampling ratios and any number of channels. Due to its scalability this sample rate converter kernel may be used both for low complexity, cost-sensitive implementations as well as for top performance applications. In a typical high performance application, sample rates of 384kHz are easily achieved on a low cost DSP and DSD input data streams are also supported for compatibility with SACD.

1. INTRODUCTION

A Sample Rate Converter (SRC) is a device or process used to transform a digital input signal at a given input sampling rate (F_{sin}) into a digital output signal at a given output sampling rate (F_{sout}).

This function is achieved using a time varying (adaptive) filter under control of a sampling rate ratio (F_{sratio}) estimator. Apart from transforming the signal's sampling rate from F_{sin} to F_{sout} , the time varying filter also needs to perform band limiting if $F_{sout} < F_{sin}$ to avoid aliasing of the higher frequency content of the input signal to the output signal.

Beside the conversion from F_{sin} to F_{sout} , a sample rate converter also provides clock domain isolation between input and output. Later in this article we shall see how a sample rate converter can be used for high performance digital to analogue conversion and how the presented SRC kernel used as a pure up-sampler provides an ideal, cost-effective solution.

A sample rate converter is said to be asynchronous if there is no simple (ratio of integers) relation between F_{sin} and F_{sout} . Throughout this document we shall assume that all sample rate conversion processes described are asynchronous.

The following diagram illustrates a generic sample rate converter [1][2]:

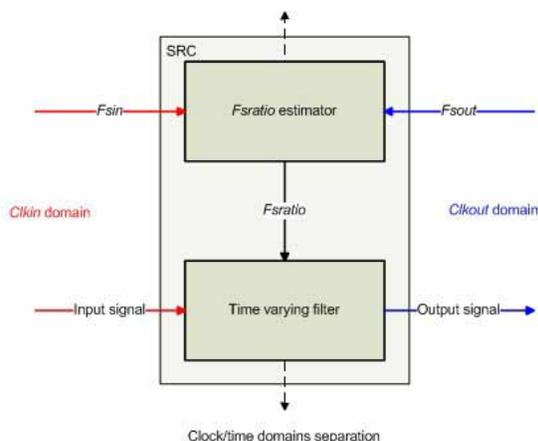


Figure 1: Generic sample rate converter

Sample rate converters are widely used to interface digital systems using different sampling rates or to synchronize multiple digital data sources to one common format. The need for efficient sample rate converters is also growing in packet based audio transmission systems where a time base needs to be recreated for playback. In this case sample rate converters may ideally replace PLLs used to generate audio clocks and to bring all incoming audio data to a common format which simplifies post-processing, analogue output stage and overall system design.

The scalable software asynchronous sample rate converter kernel presented in this paper is available from ANAGRAM Technologies¹.

2. ALGORITHM DESCRIPTION

2.1. Algorithm structure

In this section we present the new sample rate converter algorithm structure and its implementation in more details. The new kernel is not a single algorithm; it is rather a scalable and flexible algorithm structure configurable for various needs and able to provide optimal performance per available resource unit.

¹ Sample Rate Converter kernel is available from ANAGRAM Technologies under the Q5™ trademark and is patent-pending.

Unlike most commercially available sample rate converters (see for instance [3][4]), the new kernel is preferably implemented as a software library on a general purpose CPU or dedicated DSP. The software approach allows for dynamic configuration and easy customisation of the SRC process. A software based SRC has lately been presented [5].

2.1.1. Generic SRC algorithm structure

The following diagram shows a more detailed view of a generic sample rate converter:

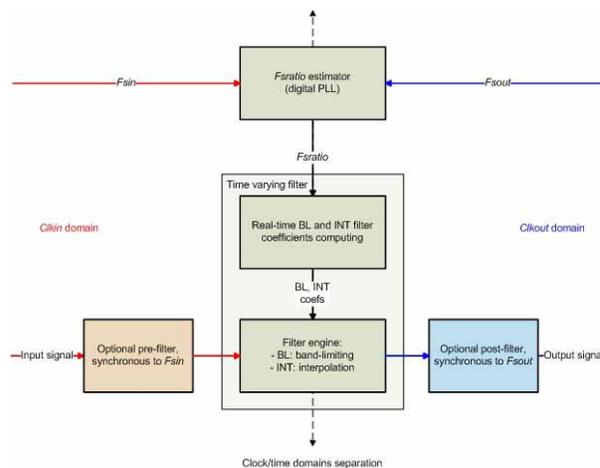


Figure 2: Detailed view of generic sample rate converter

The optional pre- and post- filters are synchronous filters providing sampling rate adaptation before and after the effective clock domain transposition stage (time varying filter).

The time varying filter is composed of two parts:

- **Interpolation part (INT):** This part effectively transposes the data sampled at F_{sin} (or at an integer multiplier/divider thereof) to data sampled at F_{cout} (or at an integer multiplier/divider thereof). It is realised using an adaptive filter tracking the relative time instants of input and output samples and providing anti-aliasing if needed.
- **Band-limiting part (BL):** This part is used to band-limit (low-pass) the spectral content of the input signal in the case of $F_{cout} < F_{sin}$. It is used in this case to avoid aliasing of the higher spectral content of the input signal into the output signal. This part needs

to be adaptive in order to accommodate different $Fsratio$ values. However, this part requires only slow adaptation as the potential changes (drifts) in input and output sampling rates are slow compared to the output signal's sampling rate.

The *BL* and *INT* stages are often combined into a single filter and the coefficients of this filter are all adapted in real time for each output sample. This may lead to high computational complexity.

2.1.2. Presented SRC algorithm structure

Band-limiting and interpolation separation

The new software SRC kernel uses an approach where the *BL* and *INT* stages are completely separated. Following diagram illustrates the structure of the proposed approach:

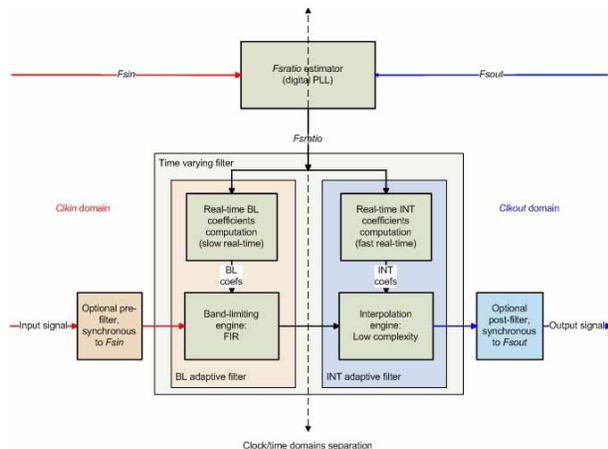


Figure 3: Structure of new SRC kernel

The separation of band-limiting and interpolation has several key advantages:

- Interpolation stage does not have to care about band-limiting the input signal; thus very simple interpolation models can be used, such as quadratic splines or Legendre polynomials. Even for very high performance a 3 taps adaptive filter is enough, which results in low computational complexity for this stage.
- Band-limiting stage needs to take care of input signal band-width reduction in case $F_{sout} < F_{sin}$. However

as F_{sin} and F_{sout} do not vary fast, the updating of the band-limiting filter coefficients is not needed for every output sample but only once in a while, resulting in low computational complexity for this stage.

- Separate *BL* and *INT* stages bring additional modularity to the system and more flexibility.

Self-configuration

Another key advantage of this separation between band-limiting and interpolation is the algorithm's ability to **self-configuration**.

If $F_{sin} \leq F_{sout}$, only a *BL* filter with pass-band up to $F_{sin}/2$ is required. As the *BL* filter is synchronous to the input signal, the *BL* filter can be implemented as a normalised filter with pass-band up to $1/2$.

If $F_{sin} > F_{sout}$, we first notice that there exists a unique positive (or zero) integer k such that $F_{sin}/2^{(k+1)} \leq F_{sout} < F_{sin}/2^k$. Thus by implementing k down-sampling by two stages as optional pre-filters, we can always get down to the case where $1/2 F_{sin} \leq F_{sout} < F_{sin}$. As such the *BL* filter only needs to be scalable for a pass-band in the $F_{sin}/4$ to $F_{sin}/2$ range [6]. Alternatively, if considered as a normalized filter, the *BL* filter's band-width must be adaptable from $1/4$ to $1/2$. We will now see how this adaptation can be performed by the proposed algorithm kernel, thus providing self-configuration.

Let's consider a low-pass filter F with frequency response $F(w)$ and band-width $[0; Fc]$.

Let r be a positive non-zero real number and consider the filter F' given by the frequency response:

$$F'(w) = F(rw) \tag{1}$$

Thus F' will have a band-width of $[0; Fc/r]$.

If we now consider BLO as a low-pass filter with normalized pass-band equal to $[0; 1/2]$ and let r vary from 1 to 2, we define $BL(w) = BLO(rw)$. The resulting pass-band for *BL* is then $[0; (1/2)/r]$.

According to the above, the sample rate conversion problem if $F_{sout} < F_{sin}$ can be reduced to the case where $1/2 F_{sin} \leq F_{sout} < F_{sin}$, which can be rewritten

as $1 < F_{sin} / F_{sout} \leq 2$. If we now consider $r = F_{sin} / F_{sout}$ and consider BLO to operate at F_{sin} sampling rate, we have, through equation (1):

$$BL(w) = BLO(r w) \quad (2)$$

And the corresponding pass-band for BL gets:

$$\begin{aligned} [0; (F_{sin} / 2) / r [&= \\ [0; (F_{sin} / 2) \times F_{sout} / F_{sin} [&= \\ [0; F_{sout} / 2 [& \end{aligned} \quad (3)$$

Thus BL provides appropriate pass-band to band-limit the input signal to $F_{sout} / 2$, which is the desired result.

On the other hand, it is well known [7], that for a filter F with frequency response $F(w)$ and time domain impulse response $f(n)$ and any positive non-zero real number r , we have, through the Fourier transform and its inverse:

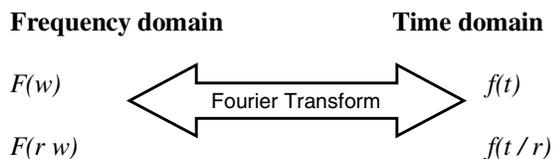


Figure 4: Time – Frequency domain relations

By applying the above to filters BLO and BL with respective time domain impulse responses $blo(t)$ and $bl(t)$, we have:

$$bl(t) = blo(t / r) \quad (4)$$

And, as $r = F_{sin} / F_{sout}$ is larger than 1, computing the time domain impulse response of BL is thus equivalent to sample rate conversion of the time domain impulse response of BLO , using an output sampling frequency of $F_{sout} = r \times F_{sin}$. As in this case $F_{sout} \geq F_{sin}$, we have transformed the band-limiting filter coefficients computation problem in the case of down-sampling ($F_{sout} < F_{sin}$) into the up-sampling problem ($F_{sout} > F_{sin}$) of a known prototype filter BLO . As such, the down-sampling case for sample rate conversion is actually configured using the sample rate conversion algorithm in the up-sampling case, thus providing **self-configuration** of the algorithm.

In terms of algorithm flow (or code size in the case of a software implementation), the same path (code) is used for the sample rate process as for its configuration, resulting in savings in complexity.

In its full implementation, the self-configuration mechanism can configure the proposed algorithm kernel for an almost unlimited range of F_{sratio} . Successful tests have been realized with F_{sratio} as large as 10^6 or as small as 10^{-6} . This is orders of magnitude more than what is supported by other commercially available sample rate converter kernels. The proposed approach may be used for arbitrary sample rate conversions, limited only by available computational and memory resources.

Remarks:

- In the case where only discrete frequency bands for input and output are of interest (as in audio applications), the BL filter configuration computations may be replaced by pre-computed filters stored in ROM for even less complexity.
- One of the key features of the presented kernel is that the architecture is able to support almost arbitrary sample rate changes, only limited by the amount of available computational resources, not by the algorithm's capabilities.
- For up-sampling only applications, the complexity of the Q5™ kernel may further be reduced as the BL filter coefficients do not need to be adapted, resulting in a very low complexity solution with very high output sampling rates supported.

3. ALGORITHM IMPLEMENTATION

3.1. General considerations

The modular, scalable nature of the presented SRC kernel and its low complexity make it an ideal candidate for software based implementations. It has been successfully implemented on a number of general purpose CPUs (such as ARM® 9xx or Pentium®) and DSP capable engines (such as Analog Devices SHARC® or Blackfin® platforms).

The modularity of the algorithm allows for easy customization to user's needs in terms of performance and MIPS / memory usage. Different implementations may be used for the interpolation and the band-limiting stages resulting in a complete family of sample rate converters covering low-end cost sensitive applications to high-end, performance oriented systems.

Another key feature of the proposed family of sample rate converter kernels is their ability to accept DSD input signals (from an SACD playback device for instance) with very little overhead when compared to a standard PCM input thanks to a dedicated DSD input module².

Being implemented as a software library, the presented SRC kernel can easily be integrated into already existing DSP resources of a given design, resulting in significant savings when compared to a hardware sample rate converter. Moreover, as a software library, the proposed SRC kernel may be upgraded in field and dynamically controlled. For instance, consider a system with given DSP/MCU resources available. The software SRC kernel may once be configured for 7.1 channels (48kHz input) when playing back a movie on DVD-V and then be configured for highest performance for stereo SACD (DSD input) reproduction, using the available resources to their best in both cases. This kind of dynamical algorithm scaling is of course not possible with a silicon based SRC implementation. It is also to be noted that as many channels as desired may be processed by the new SRC kernel and that this number can be configured dynamically at run-time.

The following table shows some typical examples of real-time SRC software implementation using the presented kernel. The band limiting prototype filter is implemented as a linear phase FIR and quadratic splines are used for the interpolation stage.

| Platform and SRC type | MIPS | Memory (excl. i/o) | THD+N (all freq.) |
|--|-------|--------------------|-------------------|
| ARM946, 2ch. PCM, 48->46.7kHz, PMP | < 24 | < 5kB | -96dB |
| SHARC21262, 6ch. PCM, 48->48kHz Home Theater | < 36 | < 10kB | -118dB |
| BF532, 2ch. DSD DSD->384kHz, Top grade DAC | < 250 | < 22kB | -145dB |

Table 1: Real world implementations

The following diagram shows the structure of the proposed SRC kernel when used as a C callable library on a general purpose MCU or DSP running an application program:

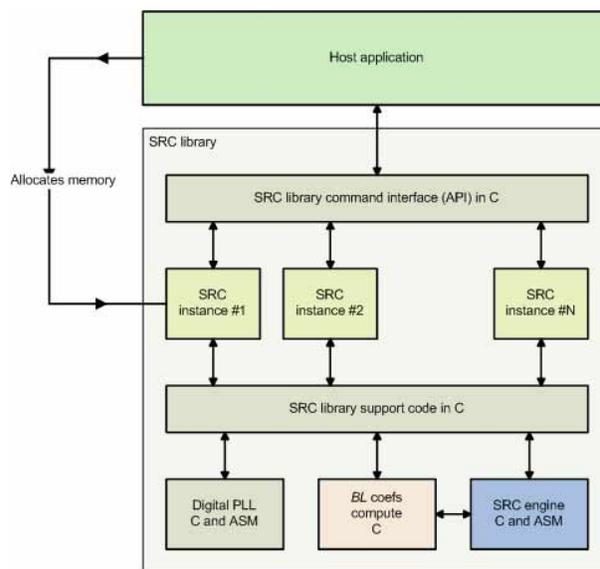


Figure 5: SRC library host interface

4. APPLICATION TO HIGH PERFORMANCE D/A CONVERSION

4.1. Reminder of modern D/A conversion

Modern D/A converters almost all rely on an over-sampled multi-bit delta sigma architectures as follows:

² DSD input module is based on ANAGRAM Technologies' DSF™ (Direct Stream Filtering) technology. Please contact author for more details



Figure 6: Modern D/A conversion chain

The first stage in the D/A converter is an up-sampler which typically brings the input signal to a sampling rate of $8x F_s$ (352.8 or 384kHz). In high performance D/A chips, this stage can usually be bypassed to directly drive the delta-sigma modulator using an external digital over-sampling filter of better performance.

The second stage is the delta-sigma modulator. This stage reduces the number of bits per sample by shaping noise into high-frequency regions. To do so, it relies on heavy over-sampling (typically up to 128x) and a shaping filter. The additional over-sampling is generally realised using sample and hold or linear interpolation with the introduced error being shaped by the noise-shaping filter.

Finally the last stage is a multi-bit DAC (usually of 5 to 6 bits resolution in high performance chips) converting the bit reduced, high-speed digital signal into analogue. A low-pass filter is needed to remove high-frequency noise introduced by the delta-sigma modulator and the quantification of the DAC.

As can be seen from the above, over-sampling is a mandatory part of modern digital to analogue converter systems. This stage can advantageously be combined with the proposed algorithm used as an asynchronous up-sampler to provide high performance, cost-effective D/A conversion solutions.

4.2. Alternative high-performance D/A conversion system

D/A conversion is the process of creating an analogue waveform from digital data. This is achieved by converting the digital code to a current or voltage, creating a time base and finally smoothing out the resulting wave form. All three of these steps are important but time base creation is often overlooked. It is however of paramount importance as jitter is one of the primary factors for audio quality. The proposed algorithm can advantageously be used to create high performance D/A converter solutions by providing a high sampling rate, jitter reduced signal to the D/A

converter chips. The following drawing shows the architectural block-diagram of such a D/A converter:

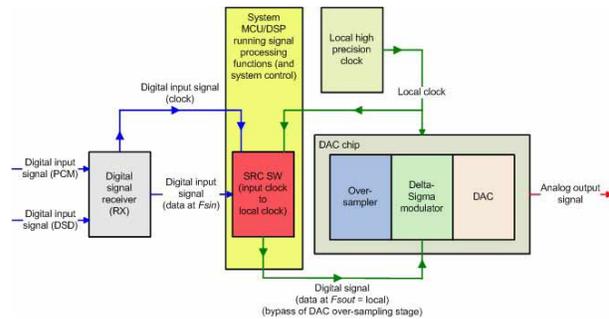


Figure 7: High performance D/A converter architecture

A real-world implementation³ of this converter architecture [8] has been realized using an Analog Devices Blackfin® DSP and AD1955 type D/A converters. The FFT of a 1kHz full-scale sine wave reproduced by this implementation is shown below.



Figure 8: D/A converter reference platform

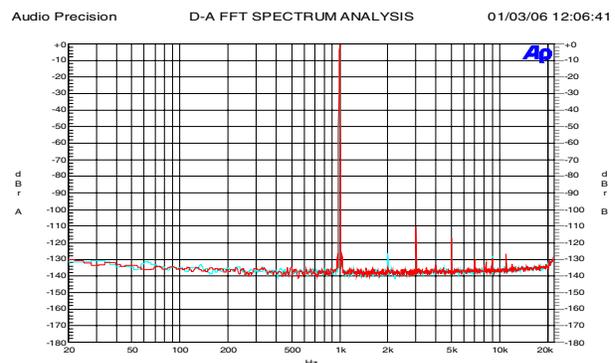


Figure 9: FFT of 1kHz full scale sine wave output

³ Sonic2™ platform from ANAGRAM Technologies

5. CONCLUSIONS

The proposed sample rate converter kernel provides an elegant, cost-effective alternative to monolithic SRC chips. Its dynamical run-time configuration capabilities make it ideally suited to a wide variety of applications requiring different sample rate conversion properties depending on context and playback content. It has successfully been implemented in wide range of consumer (and some professional) audio products ranging from cost-sensitive designs to high-end D/A converters. Its abilities to natively handle DSD input signals and high output sampling frequencies are unique in the industry.

6. ACKNOWLEDGEMENTS

This work was supported by ANAGRAM Technologies SA and was performed during the development effort of the new Sonic2™ D/A conversion platform. Special thanks go to the engineering team at ANAGRAM Technologies who transformed the theoretical concepts into real world applications.

7. REFERENCES

- [1] Udo Zölzer, “Digital Audio Signal Processing”, John Wiley & Sons, Chichester, West Sussex, England, 1997.
- [2] Thierry Heeb, “Q5™ up-sampling / sample rate conversion technology”, ANAGRAM Technologies whitepaper, 2005.
- [3] Kevin James McLaughlin and Bob Adams, “An asynchronous sample rate converter with 120dB THD+N supporting sampling rates up to 192kHz” presented at the AES 109th Convention, Los-Angeles, USA, 2000 September 22-25.
- [4] Cirrus Logic, CS8420 datasheet
- [5] Paul Beckmann, Timothy Stilson, “An efficient asynchronous sampling-rate conversion algorithm for multi-channel audio applications” presented at the AES 119th Convention, New-York, USA, 2005 October 7-10.
- [6] Patent Application B11633FR, “Procédé et dispositif de conversion de frequencys d'échantillonnage d'un signal numérique”, Thierry Heeb, ANAGRAM Technologies, 2005.
- [7] Oppenheim, A. V., and Schaffer, R. W., “Discrete-time Signal Processing”, Prentice Hall, Englewood Cliffs, New-Jersey, 1989.
- [8] Sonic2 D/A converter reference platform user manual, ANAGRAM Technologies, 2006.