Audio Engineering Society
Convention Paper 6863

Presented at the 121st Convention
2006 October 5–8 San Francisco, CA, USA

Asynchronous Sample Rate Converter for Digital Audio Amplifiers

Pallab Midya[1], Bill Roeckner [2], and Tony Schooler [3]

[1] Freescale Semiconductor, Inc., Lake Zurich, IL 60067, USA
pallab.midya@freescale.com
[2] Freescale Semiconductor, Inc., Lake Zurich, IL 60067, USA
bill.roeckner@freescale.com
[3] Motorola, Inc., Schaumburg, IL 60196, USA
anthony.schooler@motorola.com

ABSTRACT

A high performance digital audio amplifier system requires an asynchronous sample rate converter to synchronize the input digital data stream to the low jitter system clock used to generate the digital PWM output. By performing the sample rate conversion with highly oversampled signals the computation and memory requirements are minimized. The performance of the digital amplifier system is not limited by the sample rate converter while accommodating multiple input and output rates. The digital amplifier system, including the asynchronous sample rate converter, is implemented in an IC. Measured data shows linearity performance exceeding 120 dB.

1. INTRODUCTION

PWM based digital audio amplifiers convert an incoming PCM bit-stream to a digital PWM output that is then amplified by a switching power stage. The information content in the PCM amplitude is converted to time in the digital PWM output. The digital PWM waveform is generated by counting a high speed quantization clock. For high fidelity applications it is critical to maintain high accuracy in the quantization clock. The quantization clock is often generated using a local crystal clock. The input PCM signal may be based on a different crystal clock. Thus an asynchronous sample rate converter (ASRC) for converting from the input sample rate to the output sample rate is a highly desirable block in a digital amplifier system. Further, an ASRC allows the use of multiple switching frequencies to avoid interference with an AM band if the digital amplifier is being used to amplify an AM source [1]. Alternatively, the need for an ASRC is eliminated by synchronizing the PWM output frequency to the digital PCM input sample rate. However, this approach requires a PLL to provide the synchronization and restricts the PWM switching frequency.

Standalone ASRC implementations for audio signals have been documented in the literature [2]. For Nyquist-rate sample rate converters, a typical approach is to upsample the input data to a very high data rate and then to decimate it to the output rate. This approach has high accuracy but results in large computation and memory requirements. Other approaches to sample rate

conversion exist which have fidelity restrictions and are not suitable for high fidelity digital audio amplifier systems.

It is highly desirable to be able to integrate the ASRC function with the other signal processing functions of the digital amplifier. A PWM based digital audio amplifier works with highly oversampled signals. This property can be advantageously utilized to implement a computationally efficient ASRC capable of integration with a digital amplifier [3].

Starting with a highly oversampled input PCM data stream allows one to interpolate to the output sample points directly. The output sample times can be computed in terms of the input sample times using a digital servo loop. The digital servo loop calculates the relative ratio between the input and output sample frequencies. This paper describes the architecture, implementation, and results of such an ASRC integrated with a digital audio amplifier.

2. ASRC ARCHITECTURE

Figure 1 shows a block diagram of the ASRC within the digital amplifier. The input to the system is digital PCM data (for example 16-bit, 44.1 ksps CD data). An upsampler is used to produce an oversampled PCM data stream. The upsampler is chosen to have very good frequency flatness in the passband while suppressing the image frequencies by more than 100 dB. The ASRC has the function of converting the upsampled PCM data stream to another PCM data stream that is synchronized to the PWM output frequency. The PCM to digital PWM block performs predistortion [4] and integral noise shaping [5].

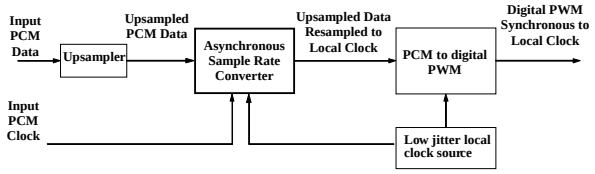

Figure 1 Digital Amplifier System including ASRC

Figure 2 shows a simplified block diagram of the ASRC. It consists of an upsampler, a digital servo loop, a buffer, and a Lagrange interpolator.

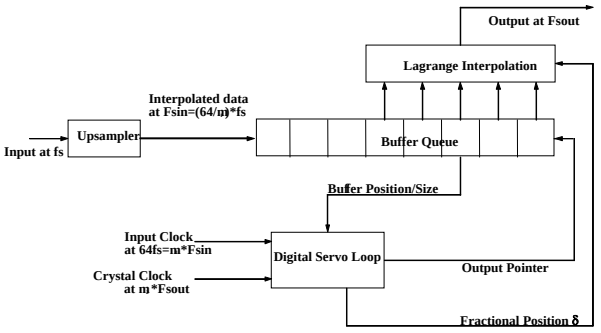

Figure 2 Simplified block diagram of the ASRC

2.1. Definition of Variables

Variable	Description
PRF	PWM Pulse Repetition Frequency
Fs	Input PCM sample rate
Fs_{in}	PCM sample rate after upsampling
Fs_{out}	PCM sample rate at ASRC output
Ts_{in}	PCM sample period after upsampling
Ts_{out}	PCM sample period at ASRC output
Fc	PWM quantization clock frequency
δ(m)	Fractional position of output sample relative to input sample
a	Ratio of Fs_{in} to Fs_{out}
u	Upsampling factor
m1	Ratio of the input PCM clock (64xFs) to the PCM sample rate after upsampling (Fs_{in})
m2	Ratio of PWM quantization clock (Fc) to PCM sample rate at ASRC output (Fs_{out})
K_1	Design constant
K_2	Design constant
K_3	Design constant
ns	Number of PWM quantization clocks (Fc) per input PCM clock (64xFs)
n	Index for input samples
m	Index for output samples
pflag	Position flag

Table 1 Definition of variables.

2.2. Upsampler

The upsampler input data is typically sampled at the Nyquist rate (Fs). There is an associated input PCM

clock with a frequency of 64xFs. The PCM to digital PWM block requires an input rate that is at twice the PWM pulse repetition rate (PRF). The upsampler is programmable to provide an upsampling factor of 4, 6, 8, 12, 16, or 24. Depending on the input PCM data rate, the upsampling factor is chosen to produce an upsampled PCM data rate that is close to the desired rate of twice the PRF, Fs_{out}.

2.3. Buffer

The buffer is a queue of upsampled PCM samples. The samples are loaded into the buffer at the upsampled rate. The samples are unloaded from the buffer at the ASRC output rate.

2.4. Digital Servo

The PCM sampling rate after the upsampler is defined to be Fs_{in}. The PWM quantization clock frequency, Fc, is a multiple of Fs_{out}. The digital servo loop operates on the fastest clocks available at both the input and output clock domains. This is 64xFs at the input rate, and Fc (which is equal to m_2xFs_{out}) at the output rate. The servo instructs the buffer to unload 0, 1 or 2 samples. It also provides the fractional time position, δ, between the output sample and the nearest input sample.

2.5. Interpolation

Figure 3 shows the input and output samples of the ASRC as a function of time. The input signals are separated by time Ts_{in} and the output samples are separated by time Ts_{out}. The two sample rates can have any arbitrary ratio. The fractional time between the output sample and the nearest input sample is defined to be δ. A Lagrange polynomial is used to interpolate the value of each output sample as a function of the input sample values and the fractional position δ. A Lagrange polynomial fits a curve which passes through "n" points. In this case n=5 has been chosen. Higher or lower orders can be chosen depending on the accuracy required. Higher orders need more computation but provide higher accuracy.

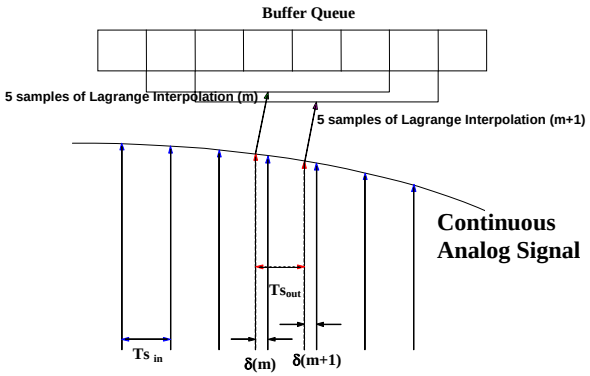

Figure 3 Input and output samples of ASRC

3. MATHEMATICAL MODEL OF THE DIGITAL SERVO LOOP

Figure 4 shows the details of the servo loop. There are three discrete time integration stages. The first integrator is a leaky integrator with leakage being controlled by the value of the coefficient, K_2. This integrator is updated every time there is an input clock (64xFs). The summation output, I, decreases with each input clock and increases with each PWM quantization clock. The required output of the digital servo loop is "a" which is the ratio of Fs_{in}/Fs_{out}. Since the system actually uses signals $m1*Fs_{in}$ and $m2*Fs_{out}$ the first integrator adds m1 and m2 for each clock signal. In the case that m1 is not an integer, an integer "p" needs to be chosen such that p*m1 and p*m2 are integers.

Using the calculated frequency ratio "a", the fractional position value δ can be obtained. The frequency ratio value "a" is added to the previous value of δ. The sum is rounded. The rounded number determines the output pointer, and the remainder becomes the new δ value.

At startup, the buffer is empty. To reach normal operation quickly, the buffer position is used to update the frequency ratio. However, under normal operation the buffer position is not used. This is controlled by a digital flag called "pflag" which is set to "1" at startup and set to "0" under normal operation.

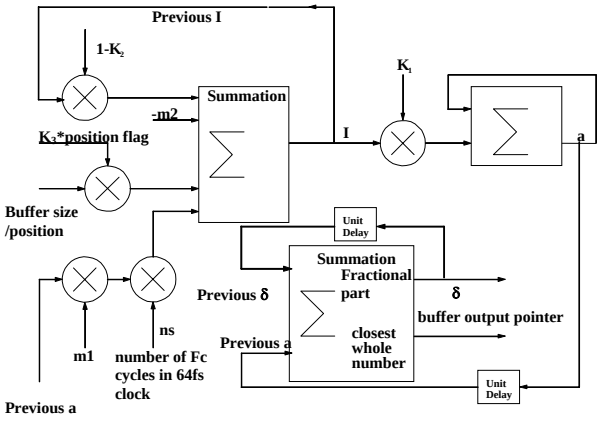

Figure 4 Details of the servo loop

The following equations are implemented in the digital servo loop.

$$a=Fs_{in} / Fs_{out}=T_{out}/T_{in}$$

$$m_1= 64xFs_{in}/(u*Fs_{in})=64/u$$

$$Fc = m_2*Fs_{out}$$

$$I(n) = (1-k_2)I(n-1)+\{a(n-1)m_1ns - m_2\}+k3.pflag.bufferposition$$

$$a(n) = a(n-1)+ k1(I(n-1))$$

$$\delta(m) =(\delta(m-1)+ a)- round(\delta(m-1)+ a)$$

$$outputpointer(m) = round(\delta(m-1)+ a)$$

4. IMPLEMENTATION

The ASRC is implemented using 24-bit fixed point hardware. The servo loop uses dedicated hardware. The coefficients for the Lagrange interpolation are computed in real time in order to reduce memory requirements. There are six independent audio channels in our implementation. The coefficients are common to all the audio channels even though the audio content between the channels is independent.

5. PERFORMANCE

The ASRC was initially simulated in Matlab, tested on an FPGA, and finally implemented in silicon. Since intermediate signals within the ASRC are not available in the IC implementation, simulation and FPGA results of these signals are shown.

5.1. Simulated performance

Figure 5 shows the error in the frequency ratio variation based on a Matlab simulation. The maximum error is less than $+/-10^{-5}$ showing the high accuracy achieved. In the IC implementation 24 bits are used for the frequency ratio. Repeated observation of the frequency ratio produces an unchanging 24-bit value.

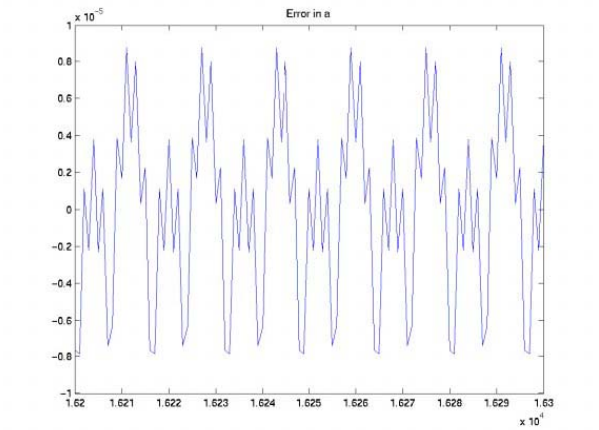

Figure 5 Frequency ratio "a" measured by the servo loop

5.2. Measured Performance of FPGA Implementation

In the FPGA implementation the PCM data at the output of the sample rate converter was captured and analyzed. Figure 6 shows the spectrum of the PCM output signal for a large amplitude 1 kHz input. This PCM signal has 24-bit resolution so the noise floor is very low. There are no observable spurious tones.

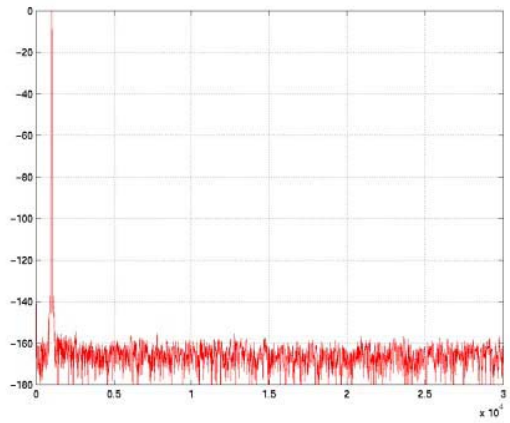

Figure 6 FPGA output spectrum with 1 kHz input

The hardest case for the ASRC is when the input signal has high amplitude and high frequency. Figure 7 shows the case when the PCM input tone is at 20 kHz and there is a large beat frequency between the input and output sample rates.

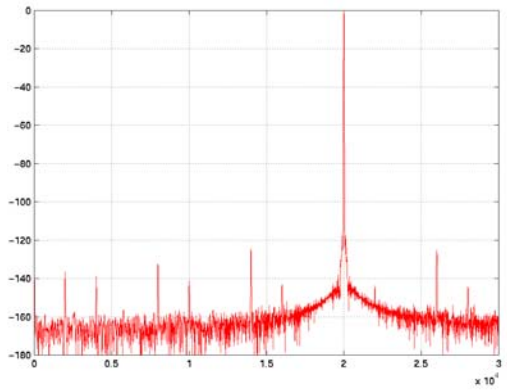

Figure 7 FPGA output spectrum with 20 kHz input

To ensure that the ASRC has acceptable performance over the entire range of operation, numerous Matlab simulations were run. Figure 8 summarizes the findings that the spurious tones would be lower than -130 dB relative to the input tone, indicating that the ASRC performance is extremely good for all the cases considered. The x-axis is the difference between the output frequency and the upsampled input frequency. Since the upsampling ratio is programmable this difference can be kept under 60kHz for all the existing standard audio sample rates.

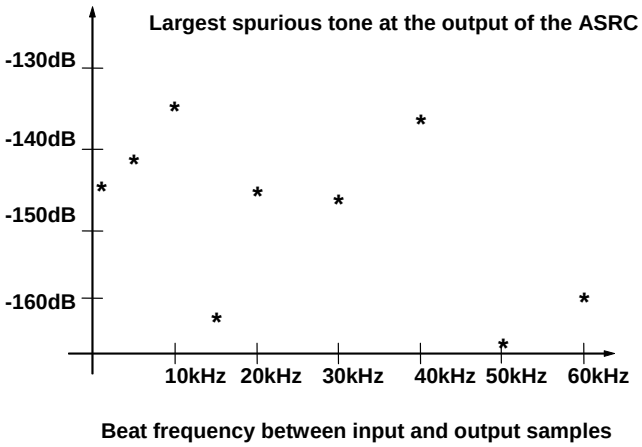

Figure 8 Matlab simulation of ASRC spurious tones

5.3. Measured Performance of IC Implementation

The performance of the digital amplifier is best measured at the output of the power stage after the LC lowpass filter. The implementation uses a high performance digital feedback loop to correct for power stage nonlinearity [6-7]. Thus the measured results include the limitations of the power stage and the feedback loop.

Figures 9 and 10 compare the output spectrum for a large signal 20 kHz tone when the beat frequency is approximately zero and 13.44 kHz respectively. The amplitude is chosen to be -8 dB (relative to the 10% THD point) so that the nonlinearity of the feedback loop

is minimized. The level of the spurious tones is less than minus 130 dBFS.

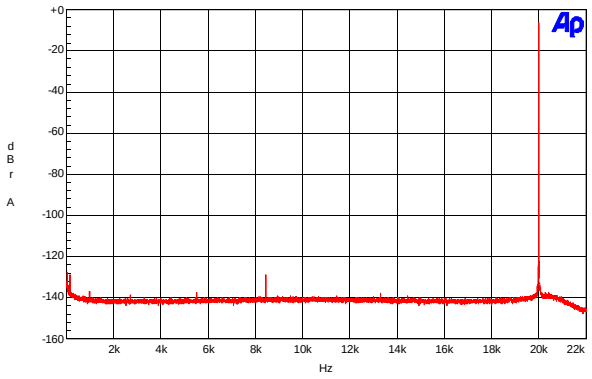

Figure 9 Spectrum at the output of the digital amplifier with a low beat frequency between input and output signals (20 kHz audio tone at -8 dB)

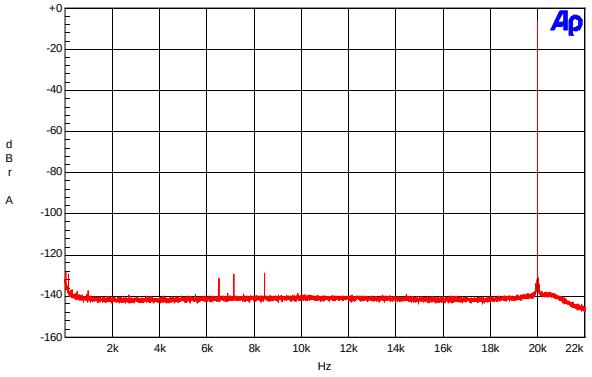

Figure 10 Spectrum at the output of the digital amplifier with a 13.44 kHz beat frequency between input and output signals (20 kHz audio tone at -8 dB)

Figures 11 and 12 compare the output spectrum for a 7 kHz large signal tone when the beat frequency is approximately zero and 13.44 kHz respectively. The level of the spurious tone is -130 dBFS. This is very low, especially relative to the second harmonic which is at -110 dBFS. At lower signal frequencies the linearity of the ASRC is even better.

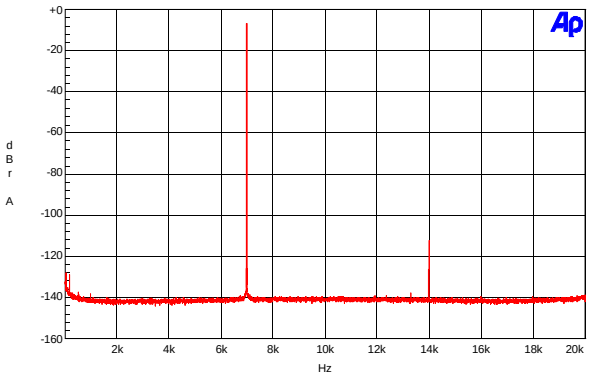

Figure 11 Spectrum at the output of the digital amplifier with a low beat frequency between input and output signals (7 kHz audio tone at -8 dB)

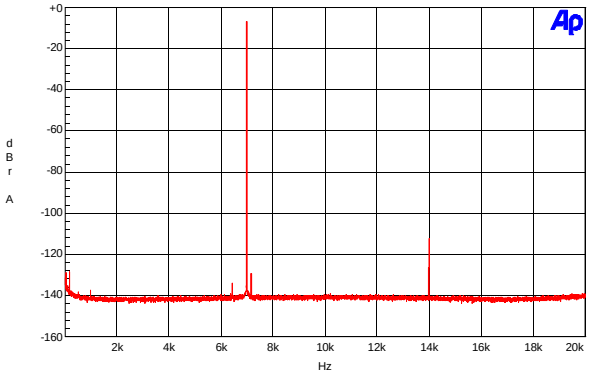

Figure 12 Spectrum at the output of the digital amplifier with a 13.44 kHz beat frequency between input and output signals (7 kHz audio tone at -8 dB)

6. CONCLUSIONS

An ASRC has been developed that is suitable for a high fidelity PWM based digital amplifier. Highly oversampled signals are used to achieve high fidelity while minimizing the memory and computation requirements. The computations associated with the ASRC are a small part of the computations required for the generation of the digital PWM signal. The spurious tones due to the sample rate converter are less than minus 130 dBFS under worst case conditions. The overall linearity of the digital amplifier is not restricted by the ASRC. Measured spectrum at the output of the digital amplifier confirms these results

7. REFERENCES

[1] W. Roeckner, P. Midya and G. Buchwald, "Variable Frequency Switching Amplifier and Method Therefore," US Patent No. 6,819,912, Nov. 16, 2004.

[2] R. Adams and T. Kwan, "A Stereo Asynchronous Digital Sample-rate Converter for Digital Audio," IEEE Journal of Solid-State Circuits, Apr. 1994 Volume: 29, Issue: 4, page(s): 481-488.

[3] P. Midya, K. K. Hicks, and A. R. Schooler, "Digital Rate Converter," US patent publication No. 20050157828, July 21, 2005.

[4] P. Midya, B. Roeckner, P. Rakers and P. Wagh, "Prediction Correction Algorithm for Natural Pulse Width Modulation," Proc. 109th AES Convention, 2000.

[5] P. Midya, M. Miller and M. Sandler, "Integral Noise Shaping for Quantization of Pulse Width Modulation," Proc. 109th AES Convention, 2000.

[6] P. Midya, B. Roeckner and S. Bergstedt, "Digital Correction of PWM Switching Amplifiers," IEEE Power Electronics Letters, vol. 2, No. 2, pp. 68 –72, June 2004.

[7] P. Midya, B. Roeckner, and T. Paulo, "High Performance Digital Feedback for PWM Digital Audio Amplifiers," Proc. 121st AES Convention, October 2006.