Audio Engineering Society

Convention Paper 6948

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

A high performance S/PDIF receiver

Paul Lesso

Wolfson Microelectronics plc, Westfield House, 26 Westfield Road, Edinburgh EH11 2QB, United Kingdom
Paul.Lesso@wolfsonmicro.com

ABSTRACT

This paper details the design and implementation of a novel S/PDIF transceiver with a very low jitter bandwidth. We describe and demonstrate a system based on multiple-loops that synchronises to the incoming data stream with a very low bandwidth and provides the original data unmodified on a clean low jitter output clock without the need for a sample rate converter. Thus we eliminate any jitter above a low frequency (typically 10Hz) on the input data and also avoid any distortion caused by sample rate converters.

1. INTRODUCTION

Most high performance S/PDIF transceivers use a phase locked loop to recover the data followed by a digital sample rate converter to resample the data onto a clean outgoing clock and hence remove any jitter. Whilst the use of sample rate converters is widespread it is not necessarily optimal [1]. Recent work has investigated the design of different filter types for audio data [2] and there is anecdotal evidence to suggest that different sample rate converters sound different.

Here we propose an alternative solution in which a fractional-N PLL is locked to an external crystal and the frequency multiplication factor of this PLL is controlled by another loop which recovers the data. Thus system detailed in this paper synchronizes to the incoming data stream with a very low bandwidth and provides the original data unmodified on a clean low jitter output clock. By using a digital control loop to adjust the speed of the fractional-N PLL we can achieve a very low jitter transfer bandwidth whilst providing a high quality clock output. Additionally the circuit requires no off chip components to make it work thus reducing costs and simplifying board level design. By avoiding the use of sample rate converters we eliminate the possibility for the audio to be distorted in this manner and provide can the originally sent audio.

Avoiding the use of large digital sample rate converters is advantageous in other ways – if the circuit is to go on the same microchip as other mixed signal components then the digital noise can degrade performance. This is particularly important for applications where the SPDIF transceiver is to be integrated on the same die as a high performance DAC.

There are also other savings to be had - reduced area for multi-channel solutions, reduced power consumption and die area. Other work has also identified jitter related issues as a major issue in S/PDIF interoperability [4,7,8].

2. ARCHITECTURE

The goal here is to synchronize a clock generated from a PLL and high quality oscillator to incoming S/PDIF data stream. This is different to usual approach of using a PLL to recover a clock from the S/PDIF data stream,

which inherently has to track the jitter to maintain lock. The architecture [3] we used is shown below.

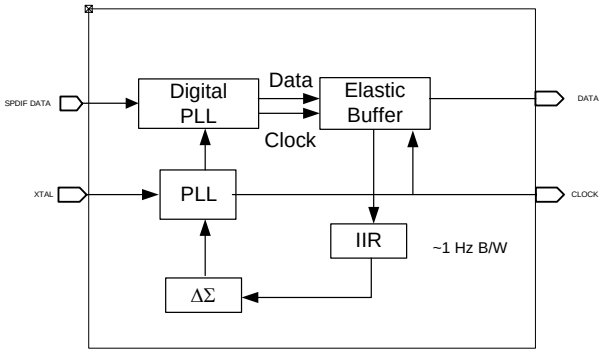

Figure 1- S/PDIF receiver architecture

The system consists of five major elements:

1. A digital phase locked loop (DPLL) that is used to lock onto the incoming data and generate an internal intermediate clock and retimed internal data.

2. An elastic buffer (EB). The EB is used to absorb any short or medium term timing variations between the local and remote clock domains. It also generates a pointer error signal indicating the current accumulated jitter or time-domain slip of the retimed internal data.

3. A digital filter (usually an IIR) that is used to attenuate the high frequency components on the pointer error signal to generate a filtered pointer error signal which drives a frequency control input of the analogue PLL. This digital filter can also be used to increase the accuracy to which the frequency error can be specified

4. A high quality clock source, such as a crystal oscillator, suitable for providing a stable time reference.

5. A high performance analogue phase locked loop (PLL) that generates a spectrally clean clock from the high quality clock source, its output frequency being controlled by the filtered pointer error, preferably acting on the feedback frequency divider of the PLL. This spectrally clean low-jitter clock is used to provide an output clock signal and to retime data out of the system.

The elastic buffer is used to determine the frequency difference between the input clock and the output clock. If the output clock is running too slowly the buffer will fill up, the pointer value will increase, and the loop will move to speed the output clock up. Conversely is the output clock is too fast then the buffer will empty and the loop will move to slow the output clock down.

This pointer will usually switch at least between two adjacent values, typically many more if there is large short-term jitter on the input data stream. It will thus have large high-frequency components, but will tend to drift up or down to track the relative frequencies of the input data clock and the clock generated by the PLL..

The DPLL output have a resolution in time of the external clock period, so has intrinsically high-frequency jitter of this order. The filter bandwidth must be chosen in such a way to ensure it can track the short-term timing variations of the jittery incoming data, but this means it also does not attenuate incoming jitter within this bandwidth. So the jitter of the recovered clock running on a 100MHz clock will typically be of the order of ten nanoseconds pk-pk.

If the jitter on this clock has strong sinusoidal tones then spurs may fall into the audio band which will degrade the quality of the audio out of the DAC. The pk-pk jitter out of the DPLL is fixed by the frequency of operation of the DPLL. The faster the DPLL is run the smaller the output jitter.

The loop is designed to attenuate jitter by adjusting the pole in the filter. To ensure data integrity any jitter which is attenuated by the loop, must be absorbed in the elastic buffer. Hence the elastic buffer and pole in the filter must be designed together.

Figure 2 shows the jitter specification for the commercial S/PDIF. The system is designed such that all the jitter over the audio-band is eliminated and absorbed by the elastic buffer.

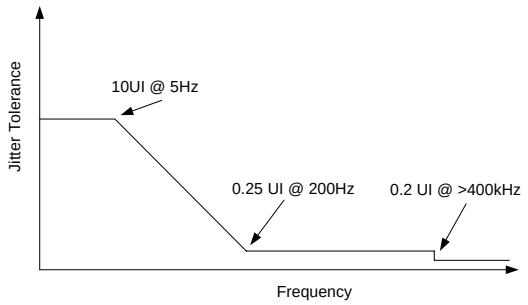

Figure 2 - Commercial jitter mask

3. AUDIBILITY OF JITTER

One of the key criteria for producing a high quality S/PDIF transceiver is the performance when connected to a DAC. There are two types of jitter that create audible artifacts in SPDIF – jitter on the data stream itself and data dependent jitter [6]. The audibility of jitter has been analyzed previously [10]. To ensure the system is immune to the effect of jitter the system is designed to have a cut-off frequency of 1Hz and all jitter above this frequency is absorbed by the elastic buffer.

4. ELASTIC BUFFER DESIGN

The function of an elastic buffer (EB) is essentially that of a buffer with asynchronous read and write interfaces. Various implementations are possible. A schematic of the EB architecture is shown below.

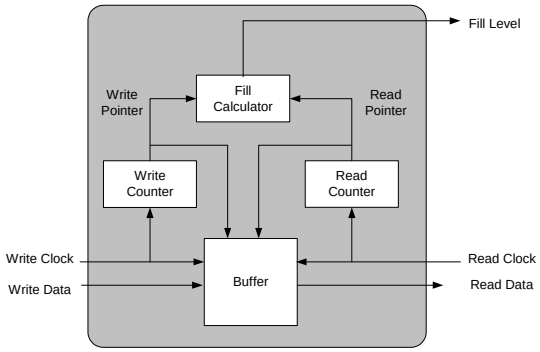

Figure 3 - Elastic Buffer

This example includes an array of storage elements. Input data are written to these elements sequentially according to an input pointer generated by a counter driven at the clock rate of the incoming data, in this case the recovered jittery clock from the DPLL. Data is read sequentially from the array according to an output pointer generated by another counter clocked at the desired output data rate, in this case the clock generated by the PLL.

The EB needs to be large enough to absorb the medium term variation in accumulated jitter that will arise from the jitter on the recovered data. EBs often find use in ensuring that data is not lost over a specific time frame. For a input frequency of R and an output frequency with an offset of Q in ppm the frequency offset is ΔR

$$\Delta R = \frac{R.Q}{10^6}$$

For a buffer of size B the time T_{slip} that can elapse before data is lost/repeated is

$$T_{slip} = \frac{B}{\Delta R}$$

Even small frequency offsets can lead to frequent data loss. For example if data is clocked into a 16-bit deep EB at 6.144Mhz and clocked out with a relative frequency offset of 1ppm, T_{slip} is just under three seconds.

Within the bandwidth of the overall control loop the jitter will be tracked but since the control loop will typically have a bandwidth of a few hertz it is essential that enough data is absorbed by the EB to cope with the jitter above this bandwidth.

The short term jitter σ_{short} is given by

$$\sigma_{short} = \frac{\sqrt{2\int_0^{\infty} S.\sin^2(\pi.\tau.f)df}}{\omega_c}$$

Where S is the single sideband power spectral density of the incoming clock and t is clock period of the sampled clock under consideration and wc is the frequency of the input clock.

For a large number of edges this reduces to give the long term jitter σ_{long} :

$$\sigma_{long} = \frac{\sqrt{2\int_0^{\infty} S.d\omega}}{\omega_c}$$

This is an r.m.s value of jitter and must be converted to a peak to peak value to determine the size of elastic buffer needed. Conversion between r.m.s. and peak to peak value of jitter for Gaussian noise is non-trivial but well known. Hence the needed elasticity in time for a specific bit error rate (BER) can be shown to be:

$$T_{elastic} = \frac{\alpha}{\omega_c}\sqrt{2\int_0^{\infty} S'.d\omega}$$

where a satisfies the equation

$$BER = \frac{erfc\left(\frac{\alpha}{2\sqrt{2}}\right)}{2}$$

Where S' is the power spectral density (PSD) of the data after passing through the DPLL minus the PSD within the bandwidth of the control loop and ω_c is the frequency of the input clock. $T_{elastic}$ must be calculated for both the reference and local clocks. There will be data loss at the specified BER if the elastic buffer can cope with pointer variations up to $T_{elastic}$ for both the local and remote clock without the pointers passing each other. For audio S/PDIF data encoded at 6.144MHz a BER of 10^{-10} corresponds to loosing a bit of data every five years. For a BER of 10^{-10}, α can be shown to be 12.723.

Hence *B*, the number of requisite bits in the elastic buffer, can be shown to be

$$B = \frac{T_{elastic\_local} + T_{elastic\_remote}}{T_s}$$

Where T_s is the data rate. This allows each pointer to shift up or down by one position for at least one clock cycle without incurring any data loss.

As the incoming clock slows down or speeds up relative to the output clock the difference between the input and output pointers will vary. The difference between the two pointers can be regarded as a pointer error signal corresponding to the number of data bits stored, which will increase if the incoming clock speeds up or decrease if the incoming clock slows down relative to the clock generated by the PLL. This output will usually switch at least between two adjacent values, typically many more if there is large short-term jitter on the input data stream. It will thus have large high-frequency components, but will tend to drift up or down to track the relative frequencies of the input data clock and the clock generated by the PLL. The pointer error signal can be regarded as a measure of the phase difference between the two clocks.

Note the size of the elastic buffer is the limiting factor on the accuracy of the frequency resolution on the output. In some cases this will be acceptable, in other cases an integrator will be needed after the EB to extend the range of the EB. The larger the size of the elastic buffer the larger the maximum error possible and hence the quicker the lock time. For low jitter systems the EB may be larger than the minimum depth of buffer needed thus ensuring correct loop dynamics.

In addition there will be an additional requirement on the size of the buffer as dictated by the jitter receive specification. Any jitter that is outside the loop bandwidth must be absorbed by the elastic buffer for the system to work. The SPDIF jitter tolerance spec is shown below. From this diagram it can be seen that the elastic buffer needs to be a minimum of 10 bits wide.

5. BENCH RESULTS

To evaluate the improved jitter rejection of this system we compared this new method to a standard method where a PLL is used to lock to the data stream directly.

5.1. Eye Diagram

A series of eye diagrams were generated for the recovered clocks from a receiver using the new technique and a receiver using a more traditional approach of clock and data recovery when a jittered input was used. The jitter on the input was applied at 1kHz from 0UI to 0.5UI in 50mUI steps. For commercial S/PDIF the specification only requires recovery at 0.2UI at 1kHz.

The red template is the clock specification from the AES3 specification. To meet the AES3 specification the eye must remain outside this red box.

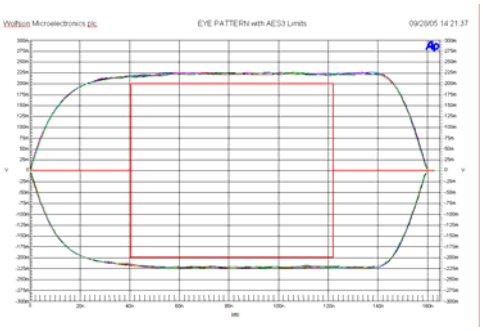

Figure 4 - Eye diagram for the clock recovered using new method

Clearly the output clock is not affected by the jitter at all. The part easily exceeds the specification for AES3.

Conversely the diagram below shows the eye diagram for a standard S/PDIF part using the standard PLL clock recovery technique. Clearly the eye closes with increasing jitter. Analysis shows the part does nothing to attenuate jitter over the audio-band and hence the recovered clock may not be good enough to run high performance converters.

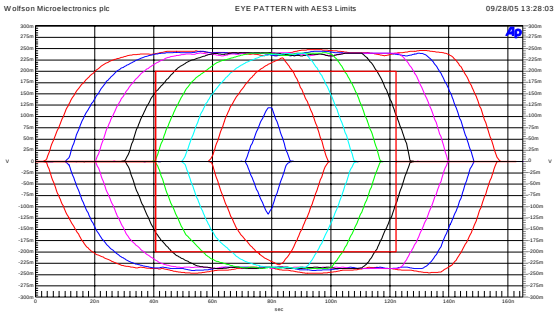

Figure 5- Eye diagram for clock recovered using standard PLL methods

5.2. Output clock histograms

Whilst period jitter is a bad metric for predicting the performance degradation of an audio converter [5] , the histograms used to calculate period jitter can be instructive. 5UI of jitter is applied at 1kHz to the input of both S/PDIF receivers.

The results for part using the new clock recovery method are shown below. The calculated period jitter is 50ps rms (the same as for no jitter applied to the part) and the histogram shows a largely Gaussian shape indicating little in the way of spectral tones.

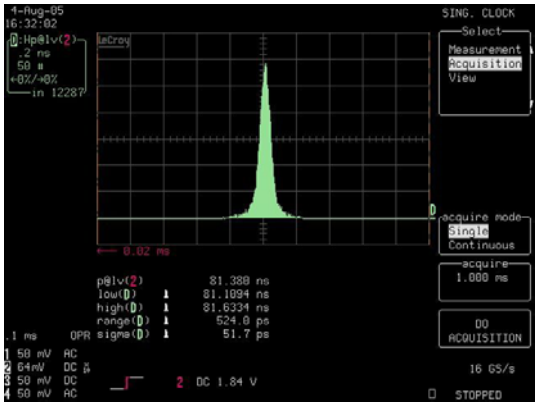

Figure 6 - Period histogram from new method with 5UI of jitter at 1Khz

Conversely the part using the standard technique is shown below. The scope does not distinguish between Random jitter (RJ) and deterministic jitter (DJ) and hence states an erroneous value for period jitter of 334ps. Clearly the histogram shows a strong sinusoidal DJ component (at 1kHz) and a RJ component in addition to that. The DJ component varies strongly with input jitter showing the output of this part to be heavily dependant on the amount of jitter on the input.

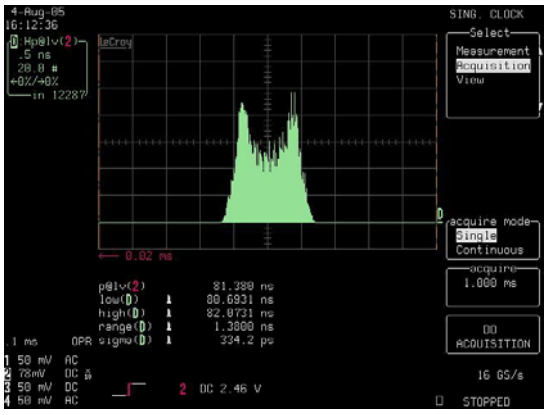

Figure 7 - Period histogram from traditional method with 5UI of jitter at 1Khz

6. ACKNOWLEDGMENTS

The author would like the thank Chris Travis for many useful discussion on jitter and S/PDIF issues.

7. CONCLUSIONS

This paper has described a novel S/PDIF receiver, which achieves the goal of a low jitter output clock and a high attenuation of jitter on the input data stream. This is significant since jitter is largely seen as a major hurdle in S/PDIF operation.

8. REFERENCES

[1] J. Dunn, "The AES3 and IEC60958 Digital Interface", Audio precision technote 26

[2] P. G. Craven," Antialias Filters and System Transient Response at High Sample Rates", Journal of Audio Engineering Society, Vol. 52, No.3, 2004.

[3] US Patent application 2005-0220240 "Clock Synchronizer and clock and data recovery apparatus and method", P. Lesso, October 2005

[4] Frandsen, Christian; Lave, Morten; "Plug and Play? An investigation into problems and solutions of digital audio networks"; AES Convention Paper, May 2004.

[5] C. Travis and P. Lesso, "Specifying the Jitter Performance of Audio Components", AES 117th Convention, October 2004

[6] Julian Dunn, "Jitter Theory", Audio Precision Technote 23

[7] Julian Dunn and Ian Dennis, "The Diagnosis and solution of jitter-related problems in digital audio systems", AES 96th Convention

[8] C Dunn and M Hawksford, "Is the AESBU/SPDIF Digital Audio Interface flawed?", AES 93rd Convention

[9] N Da Dalt, M Harteneck, C Sander and A Wiesbauer, "On the Jitter requirements of the sampling clock for analog-to-digital Converters", IEEE transactions on Circuits and systems-1, Vol. 49, No. 9, September 2002

[10] E. Benjamin and B. Gannon, "Theoretical and Audible effects of jitter on Digital Audio Quality", AES convention paper, September 1998