

How can we get rid of jitter?

Now this torture has to end. What lies in our hands in order to hear or record some music without or with less jitter?

Principally there are three approaches:

1) Use the best equipment available.

This approach is extremely expensive.

Use the newest and most sophisticated audio equipment and rely on the manufacturer that provisions against jitter have been provided.

Use the best digital interconnects:

Use SToptical instead of Toslink.

Use AES3 interconnects (XLR) instead of S/PDIF (RCA).

2) Use that audio equipment that you already have, and attenuate the jitter at specific points of your audio installation.

This approach is very cost effective.

Use external "jitter attenuation devices" prior to:

AD conversion

DA conversion

Real time sample rate conversion

3) Use the best equipment available, and do additional jitter attenuation at dedicated points of your installation.

This is a combination of 1) and 2). Install high-end equipment and use additional jitter attenuation devices prior to AD, DA, and SR conversions.

Avoiding jitter in digital recordings:

Avoiding jitter is a must for digital recording (analog to digital conversion), because if the music is recorded together with jitter, its too late, the original or potential quality is gone, forever.

The first thing that has to be accomplished is that the clock signal that controls the AD converter has a significant quality. Having a precision clock source in the studio is only one step, the other is to ensure, that the clock is properly distributed, that it arrives at a high quality level at the clock input of the AD converter.

The second thing is to avoid sample rate conversions whenever possible. When real time sample rate conversion has to be done, be sure, that the sample rate converter has very stable (low jitter) input and output clocks.

(Sample rate conversion can also be done "offline" with a personal computer and dedicated software. However, this article will treat only real time sample rate conversion with ASRC chips.)

Attenuating jitter

In a digital transmission we will always have jitter because we never have infinite precision. The goal is to attenuate the jitter to a value, that is so small, that it wont bother us.

Different approaches exist to attenuate jitter in a digital transmission line.

Sample Rate Converters

"In a perfect digital world, there would be one sampling frequency. Unfortunately the world is populated by many different sampling frequencies" says Ken C. Pohlman in his book: "Principles of Digital Audio" .

An ASRC (asynchronous sample rate converter) such as the [AD1890](#) (go to the datasheet.pdf) from [Analog Devices](#) (a semiconductor manufacturer) is a great tool for converting sample rates.

If it had not been invented you would have to do a DA conversion and an AD reconversion with the required sample rate in order to accomplish a sample rate conversion. This would heavily degenerate your signal quality.

With a digital ASRC this task will introduce less degeneration, since the signal is processed entirely in the digital domain.

However, a sample rate conversion is not a lossless process and the degree of signal-quality degeneration depends heavily on the amount of jitter at the input and output clocks.

If you use an ASRC as a jitter attenuation device, the jitter at the input of the ASRC will be distributed into the output signal-data, and what was simple clock-jitter at the beginnig is now forever glued your digital audio signal, it has become something comparable to sampling jitter.

(I have to add here, that the designers of ASRC chips try to minimize the susceptibility to input clock jitter, and this is the reason why manufacturers use ASRCs in order to reduce jitter. It may sound a little better if you add an ASRC but the price you pay, is to loose the original sound quality that **was contained** in the input signal to the ASRC.)

The clock jitter at the output of the ASRC might be less, but the signal is not the same (the data has been altered), it now irrevocably "contains the input jitter" and the initial signal quality is degenerated.

Unfortunately **sample rate converters are praised as jitter attenuation devices** by some manufacturers, **but they are not.**

ASRCs are sometimes contained in DA converters where they convert the input signal to the maximum sample rate of the converter, because manufacturers believe to get a better sound, if the output sample rate is higher.

Some manufacturers of DA converters state, they have a max. jitter amplitude of say 5 picoseconds at the converter chip. This is fine. But if this DA converter has an internal ASRC, forget the 5 picoseconds. It says nothing about the sound quality concerning jitter.

Other manufacturers implement ASRCs in their CD transports as a means of jitter reduction. What happens is, that the jitter that is generated by the transport is glued into the digital output signal by the ASRC. The output signal of such a CD transport may have low jitter, but the signal does not carry the exact data, that is on the CD. The signal quality is by definition degenerated. Funny enough, that you find this topology in quite expensive CD transports whereas the cheapest CD-players are able to digitally output the exact CD-data (with jitter, but original data).

To say it in two words:

Use sample rate converters for converting sample rates, not for attenuating jitter.

If you want to convert sample rates with high quality, use low-jitter clocks at the input and the output of the ASRC.

Repairing a jittered recording:

Now what does mean ? Didn't I say that if a recording is made with a poor clock its too late to get it right again ? **Yes.**

And didn't I say that asynchronous sample rate conversion is bad ? **Yes.**

However I found out, that the sound quality of a jittered recording can be greatly improved when the signal is run through a [JISCO](#) and subsequent ASRC.

How is this possible ?

The [JISCO](#) will scramble the jitter of the source recording and shift it into the MHz region while then the ASRC recalculates the recording and thus removes the original jitter correlations that made the original recording sound bad.

This all happens in the digital domain.

The process averages out the timing errors of the original jittered recording and can almost restore it to the quality-level as if recorded with perfect clock.

What a perfect world :-)

(Please note that there is no need and no benefit in applying the above process to a non-jittered recording.)

PLL based solutions

Jitter is best attenuated with phase locked loop (PLL) techniques.

These do not alter the signal data, but are able to correct the signal timing.

A phase locked loop is a device, that can follow the frequency of its input signal and generates an output signal of the same frequency.

The trick is, that the selfgenerated frequency is cleaner, than the frequency of the input signal.

To accomplish this, a PLL consists of a voltage controlled oscillator (VCO) which generates the frequency of the output signal, a phase detector (that compares the input frequency with the output frequency) and a loop filter that is connected between the output of the phase comparator and the input of the VCO and provides that the output frequency changes slower, than the input frequency.

Or it may be better to say: the PLL tries to align the phase between the input and output clock and by attempting this, the same frequency is automatically generated.

If you want to learn some basics about phase locked loops, look into "The Art of Electronics".

Or take a look at one of the most popular PLL chips, the [4046](#) ([go to the datasheet.pdf](#)).

Since a S/PDIF or AES signal is biphase mark encoded, it carries the clock information within. This clock information can be extracted with a PLL circuit and used to reclock the data.

This process is called **clock recovery and data retiming**.

If we use a VCXO (voltage controlled crystal oscillator) instead of a VCO as a frequency generation device in the PLL circuit, we have a very stable frequency source with low jitter.

The tradeoff is, that it is much more expensive than a VCO and that the frequency control range is very narrow, typically some hundred ppm (parts per million).

That means, that with one VCXO you can receive one sampling frequency. If you design your VCXO based PLL for 44,1kHz, it cannot be used for DAT (48kHz) or DVD (96kHz) applications. This is one of the reasons, why VCXOs are seldom used in DA converters.

There must be multiple VCXOs, one for each supported sampling frequency, and the sampling frequency must be detected automatically in order to select the appropriate VCXO for clock recovery. Varispeed applications are also not possible with VCXOs. For this reason some manufacturers use an ASRC for "jitter attenuation", because this works with all common sampling frequencies (including varispeed) although the achievable performance is considerably less, than with the VCXO based PLL approach.

Jitter attenuation can be accomplished with single-stage or dual-stage clock recovery.

Single stage clock recovery means that there is one single PLL, that is used to recover the clock, attenuate the jitter, and retime the data. A single stage clock recovery circuit can be designed with high jitter attenuation performance at a reasonable price.

Dual stage clock recovery uses two separate PLL circuits:

The first PLL does only clock recovery. The recovered clock is input into the second PLL, that is used for jitter attenuation and data retiming.

Dual stage clock recovery has the advantage, that the second PLL receives a clock signal (instead of a biphase mark signal) that is already cleaned by the first PLL.

With dual stage clock recovery, a higher jitter attenuation is attainable, and the residual jitter can be more decorrelated from the input data stream. However dual stage often means dual price.

You find more detailed information about clock recovery in Dunn's ([of Nanophon](#)) article: "Towards common specifications for digital audio interface jitter" **Βλέπε jitter_5.pdf**.

Dual stage Clock recovery can also be combined with a **data buffer memory**.

The first PLL writes to the buffer, and the second PLL reads from the buffer and tries to keep it always half filled.

If we use a FIFO (first in first out) memory as a data buffer, we have more time to react to clock variations of the input signal and thus are able to attenuate lower jitter frequencies. The larger the buffer, the lower the jitter frequency that can be attenuated.

The disadvantage of the FIFO memory solution is (beside its higher price) that the output data will be delayed (the buffer has to be half filled before data is output).

Clock recovery systems without data memory can have a delay as low as 1/2 bit period (f.e. 90 nanoseconds @ 44,1kHz sampling rate). This delay is fixed and therefore memoryless jitter attenuation devices can be used synchronously for multiple channels.

Memory based devices have a variable delay and are less suited for digital recording and broadcasting applications.

I am speaking of external jitter attenuation devices. Of course, memory based jitter attenuation can be found inside every CD-, DVD- or DAT-transport etc.

Text (c) by [Charles Altmann](#)