

Digital audio transmission, SPDIF, and Jitter

Jitter

Consider a stream of samples at 44.1 kHz with 16-bit precision. This means we have 65536 different amplitude levels. Now, consider a simplistic non-oversampling DAC which outputs a constant signal, at the sample level, during each sample. Consider a signal which is entirely made of null samples with just one spike at maximum amplitude. The DAC will output this as a rectangular pulse, $1/44100$ s in width.

In order to correctly reproduce this signal, we must have as much precision in the time domain as we have in the amplitude domain.

The SONET standard states that "Jitter is defined as the short-term variations of a digital signal's significant instants from their ideal positions in time. Significant instants could be (for example) the optimum sampling instants". The Fiber Channel standard simply defines jitter as "The deviation from the ideal timing of an event."

In short, the term "jitter" describes timing errors within a system.

Jitter is defined as clock noise : it is the RMS value of the difference between the real clock edge time and the ideal one. An ideal clock will have no jitter but a real-world one will always have some.

What precision do we need ? A quick calculation tells us that : maximum allowable jitter in order not to lose bits of precision should be about $1/(F_s \cdot 2^{\text{Bits}})$. In our case, $1/(44100 \cdot 65536) = 346$ picoseconds.

This figure should not be regarded as too accurate, but it gives us an idea.

The effect of jitter is difficult to quantify, but its presence gives the impression of loss of instrument definition, narrowing of stereo image, collapse of soundstage depth, etc.

Jitter is more apparent in higher frequencies than lower frequencies. If it is periodic rather than random, it has an effect similar to 'flutter.' For a given test tone, it will produce sidebands equal to the jitter frequency. While the level of this distortion is low, it can still be a significant artifact.

For more information, read this [interesting article](#), featuring jitter audibility curves (which are much more severe than my rough estimation).

The Clock is Analog

While the clock may look like a digital signal (it comes, after all, from digital circuitry and has only two levels), it is in fact an analog signal.

A digital signal is defined as a signal whose meaning comes from its state (0 or 1) at a particular time (say, when it is latched in a circuit). Digital signals are very noise-proof because it takes a lot of noise to fake a digital level (change a 1 into a 0 for instance).

An analog signal is defined as a signal whose meaning comes from its amplitude and/or the variation of such amplitude over time. Depending on how the valuable information they carry is encoded, analog signals can be anything from relatively immune to noise (think about an FM radio signal) to extremely sensitive (think about an analog audio signal).

The clock signal carries its information not in the logic levels but in the precise timing of their transitions (aka. "significant instants"). In this sense, it is an analog signal, as the timing information is defined by comparing the analog value of the clock signal with a fixed threshold and marking a clock tick every time the value crosses the threshold. Hence, it is very sensitive to noise and other alterations :

- Noise added to the clock will shift the transitions.
- Low-pass action by cables and other circuitry will soften the edges and render them more vulnerable to noise.
- Low-pass filtering will allow other signals (like data) to contaminate the clock.

Thus, if it is to respect the extremely strict requirements we must put on it regarding Jitter, the clock becomes a very fragile signal indeed. It must be handled with a lot of care.

Enter SPDIF.

What is SPDIF

It simply is a standard for encoding digital audio in the form of a single signal with the following properties :

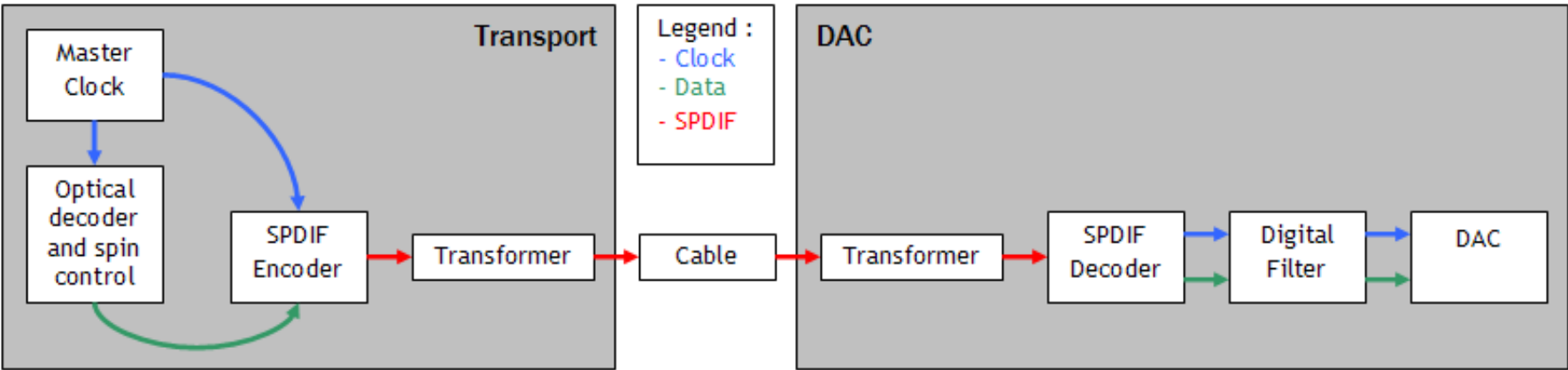
- No DC component (bandwidth-limited)
- Differential (no ground)
- Single Signal (mixed clock and data).

This signal can then pass various bandwidth-limited transmission media, especially insulation transformers, which are used to break ground loops.

SPDIF is good at transmitting amplitude information : it can always recover error-free bits and a perfectly usable bit clock. But this bit clock is not clean enough to be used as a sample clock. It has way too much jitter.

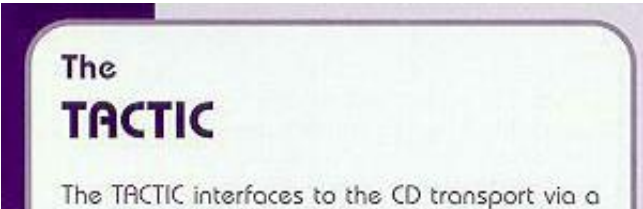
Why is SPDIF unusable ?

Consider the path the clock has to go through in a traditional Transport-cable-DAC configuration :

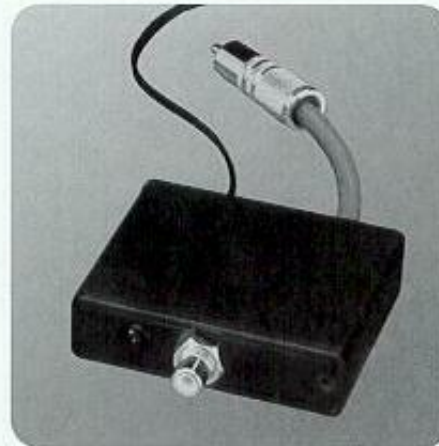


This path is very long and has plenty of opportunities for the clock to get polluted. First, the transport master clock is generally full of jitter. Second, all the lowpass filtering occurring in the cables and insulation transformers will happily smear the edges and mix them with random data levels noise. Finally, the most perilous step of all is clock recovery in the SPDIF receiver. These are not specified for low jitter. Moreover, SPDIF chose a coding system that renders it impossible to recover a clock without random data noise in it. Incredible indeed !

SPDIF is so bad it has given birth to its own Snake Oil ecosystem. Here is a monument to it :



4 inch captive input cable. The digital interconnect connects its output jack to the D-to-A converter or to the input of the AUDIT if it is used. The TACTIC's design is based on a proprietary pulse transformer circuit developed by Audient Technologies. The TACTIC isolates ground loops and absorbs noise outside the frequency bandwidth required for digital data transmission. Impedance matching networks provide critical impedance loading for the digital cable to suppress signal reflections.



TACTIC Transmission Aperture Specifications:

Dimensions: 2.8" d x 2.8" w x 0.8" h with
2.2" x 1.8" x 1.6" wall mount
transformer

Input Impedance: 75 ohms +/- 0.5 ohms
(110 +/- 1 ohm for AES/EBU version)

Output Impedance: 75 ohms +/- 0.5 ohms
(110 +/- 1 ohm for AES/EBU version)

Bandwidth: 75kHz to greater than 50 MHz

Power consumption: Less than 1 watt

No wonder different transports, cables, etc. influence the sound of the DAC. They all have their own jitter signature. Bits are bits, and they are always decoded correctly (unless a catastrophic error occurs, but this can be noticed as a very big clicking noise). People have rambled about different error correction schemes used in transports, and many other funny ideas, but I think in truth Jitter explains the transport differences.

Note that I did not say that digital processing was not important. Badly designed digital filters can wreck the sound. But these are in DACs not in transports.

For more information, read this [interesting page by Norman Tracy](#).

SPDIF and AES-EBU

AES-EBU uses the same encoding as SPDIF, but the transmission is differential instead of single-ended, and cable impedance is different too.

In practice, many CD players have a transformer at their SPDIF output and many DACS have one at their SPDIF input too. So, the signal ends up being balanced anyway, and there are no ground loops. What is the difference then ?

AES/EBU has 3 wires (2 for signal, 1 for ground) whereas SPDIF has only 2 wires (signal and ground). Thus AES/EBU emits less noise (because the cable shield is grounded and does not carry any signal), whereas noise emission in SPDIF could be important if the cable shield is disconnected at both ends in order to prevent ground loops.

What to do with SPDIF ?

Some have found other solutions :

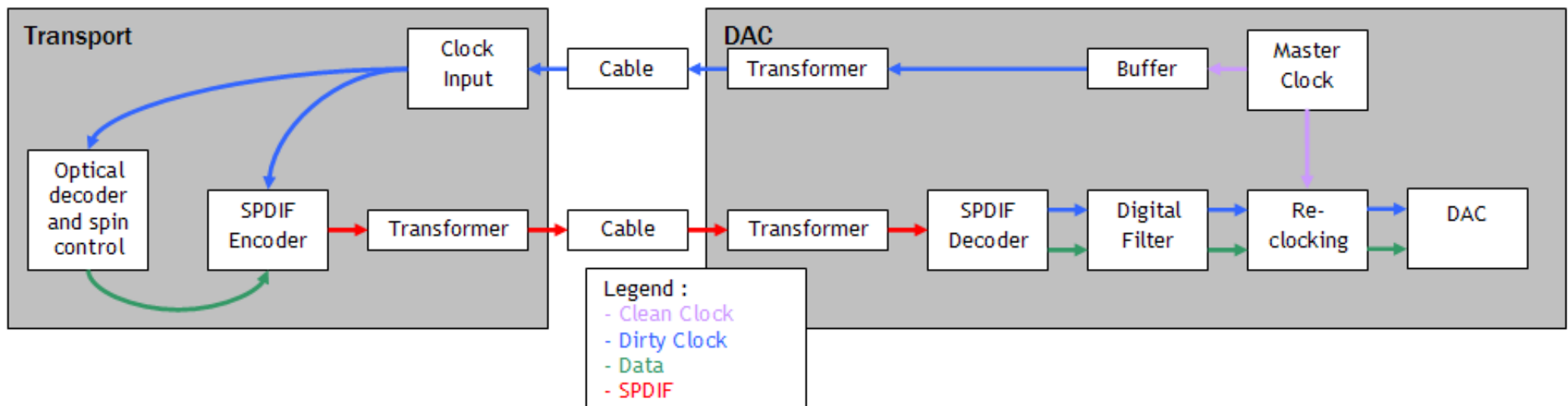
- Design a clock recovery PLL with very low jitter and recover a clean clock from SPDIF.
To date, only one project I know of has done this successfully, it is the DAC designed by the Guido Tent team. While their solution is very elegant and beautifully designed (and seems to work), I still think it is very complex.
That it took a team of very competent engineers a few years to design a PLL that could make SPDIF usable for its original purpose is a testament to the inadequacies of this transmission format.
- Keep the master clock in the transport and use a separate clock line.
Data can be transmitted as SPDIF but also raw, in which case you need 3 lines (this is I2S transmission). IMHO this is a flawed idea as transmitting the clock over a run of cable will inevitably pollute it. Besides, we might as well go all the way into incompatibility and :
- Put the clock in the DAC and send it back to the player (clock injection, see below).
- Use a transmission format which works. Sony did just that for DSD, they used FireWire. The clock is in the DAC, and it asks data from the transport as it needs it via the FireWire link, which is bidirectional. This is the really intelligent approach, unfortunately it is not accessible to DIY maniacs.

Clock injection

I like simplicity, even if this means drilling a hole in my CD player. Clock injection is simple.

Find the transport master clock. Desolder it, drill a hole in the case, and fit a BNC clock input socket, connected with appropriate coaxial cable and correct impedance termination.

Then, the DAC must send a clock signal of the appropriate frequency on this clock input to feed the transport. All systems share the same clock and are automatically synchronized. SPDIF is still used as a means of transporting bits between transport and DAC (because I2S can't go through an insulation transformer), but SPDIF does not carry the clock anymore. Here is a schematic to explain it :



Conclusion

On the diyaudio.com forums, I saw this topic on the sonic signature of SPDIF isolation transformers :

Hey, I'm not saying that different pulse transformers sound different in your DAC. But it is possible to make the problem irrelevant by changing the DAC architecture.

And some guru responds, quite contemptuously : Great to hear it, yet another breakthrough...care to 'splain how?

So most people don't want to understand this problem and will continue wallowing about the difference in sound between pulse transformers, while it is possible to sidestep the problem entirely by using clock injection.

[Next page](#)

[Back to main page](#)