March 1, 2020

Understanding how perturbations on digital signals can affect sound quality without changing bits, and how these issues are addressed by the UpTone EtherREGEN.

By John R. Swenson

Although we are presenting this white paper in the context of the UpTone EtherREGEN switch, much of the following information applies not just to Ethernet, but also to any digital audio data delivery system—Ethernet, USB, I^2S, S/PDIF et cetera. We'll work backwards from the DAC (Digital Analog Converter) to upstream interfaces.

Much of perceived digital audio performance is dependent upon the quality of the clock signal entering the DAC chip (or discrete DAC ladders) and the presence of jitter. Jitter is any variation in timing of the clock from a mathematically perfect timing reference. Jitter causes distortions in the audio signal coming out of the DAC. Exactly what those distortions look like depends upon the characteristics of the jitter components.

Clock thresholds, ground-plane noise, and receiver jitter:

So where does jitter come from? All oscillators (which generated a clock signal) have jitter, some more, some less. We classify this jitter component as "source intrinsic jitter." Additional jitter can be added to this at the receiver of the clock, which is typically a flip-flop inside the DAC chip. The flip-flop has its own intrinsic jitter, which is called "receiver intrinsic jitter." But there is a third class, "threshold jitter," which is harder to conceptually understand, and threshold jitter is what will be discussed in the most depth in this paper.

Even though we tend to think of digital signals as being "binary" with just two possible states—high (1) and low (0)—changing instantaneously between them, this is not actually the case. The signal is an analog voltage. When going from a high to a low state it hits all the intervening voltages, and the change takes a finite interval of time. This is called the "ramp time,"—the time it takes to "ramp up" or "ramp down." This time interval can be very fast, but it is still not zero. A common ramp time found in modern digital audio circuits is several nanoseconds (ns). A ns is one-billionth of a second, and while that sounds fast, in the realm we are discussing it is not. This is important because every circuit receiving a clock signal has a threshold voltage. This is the voltage above which the circuit thinks the signal is a 1, and below which it thinks it is a 0. So what happens when this threshold changes? Think of a line going up from left to right, with a threshold at say 2 volts. Whenever the signal gets above 2V, the circuit sees it as going from 0 to 1. But what if the threshold changes from 2.0V to 2.1V? The signal takes a little while longer to get to 2.1 than 2.0, so the time it takes the circuit to see this change to a "1" also increases slightly. Note that the clock signal itself did not change; *the change in timing was entirely a property of the receiver.*

Okay, so why is the receiver changing threshold? Well the receiver itself really isn't, it stays at 2.0V. Where does the extra 0.1V come from? **The voltage on the ground pin of the receiver!** The 2.0V is not some physical property of the universe. It is the voltage between the ground pin and the clock input. However, if there is any voltage between the ground pin of the receiver and the ground pin of the source, this difference will be seen as an effective change in threshold, thus changing the timing the receiver sees, otherwise known as threshold jitter.

This is the underlying mechanism for threshold jitter: *Noise on the ground-plane of the DAC causes effective jitter in the DAC chip, which modulates the audio signal coming out of the DAC.* Note that the actual clock signal does not change at all. It is how the clock circuit *inside* the DAC chip sees the clock signal that matters.

Sources of ground-plane noise

We have identified two primary external mechanisms—present in various digital interfaces—which can cause ground-plane noise on a DAC's internal ground-plane: 1) Jitter carried through digital data and, 2) leakage current.

1) Jitter carried through digital data:

We are going to explain how jitter of the clock—e.g., in a network switch—will show up in the data coming out of that switch, even though no separate clock signal is transmitted with the data. And we will explain how this gets into and affects the DAC-attached endpoint.

The critical part is what happens inside any Ethernet receiver (known as a PHY, physical interface chip). This takes some understanding of how digital circuits work: we have a wire with voltages going up and down that correspond to a digital signal. When the signal voltage goes above or below the threshold voltage of the circuit, it causes the circuit to "change state." The process of doing this creates a little "burp" of current to flow between the power and ground pins of the circuit. So every time the signal changes, small current pulses flow through the ground-plane. Remember, the timing of the data changes are controlled by the clock in the switch, so they have the phase-noise characteristic of the clock in the switch. As a result, the pulses happening in the ground-plane *also* have the same timing characteristics as the clock in the switch!

To understand this, let's back up a bit further and discuss phase-noise:

Phase-noise is an alternative way to look at jitter. You can think of it as the "frequency spectrum" of jitter. Phase-noise is usually thought of as only a property of clocks, but this is an *important noise component* because the clocks are used to determine exactly when "edges" happen in data signals. The data is "clocked out" by the clock, so timing variations in the clock, (jitter) show up in the data signals.

Explaining phase-noise is probably easier graphically or with an animation, but we will attempt it with words. The following example, while not strictly accurate, will suffice as a basis to understand how this works. Let's consider a 10MHz clock signal: there is supposed to be exactly 100ns between each rising edge of this clock signal. In reality it is never exactly perfect, it might be slightly more than 100ns, or slightly less. Think of a clock where this "error" in timing changes in a smooth and repeatable way. Let's say one pulse to the next is exactly 100ns, then the next is 101ns, then the next 102, then 103, then it goes back to 102, then 101, then 100, then 99, then 98, then 97, then 98, then 99, then 100, etc. This timing error pattern will take a certain amount of time to complete. If say it takes $1/10^{th}$ of a second for this pattern to occur, the frequency of the clock modulation is 10Hz. This is usually called jitter frequency. If you made a phase-noise graph of this clock, you would have a spike at 10Hz.

In reality, it is rarely just one frequency but many frequencies. And while it may look like noise, this is NOT an analog voltage noise such as tape hiss. Rather it is comprised of a large set of jitter frequencies. All phase-noise plots of real oscillators are not flat; the low frequency range has much higher phase-noise error, tapering off at the higher frequency range. Thus the jitter is composed primarily of lower frequency

components. If you look at the noise on the ground-plane, you will see the same low frequency dominated curve as the clock phase-noise plot of the clock in the switch.

Clock phase-noise overlay:

Now it becomes more complicated, because there is a *different* clock in the endpoint—with its own phase-noise, switching other circuits in the endpoint—and creating ground-plane noise correlating to the phase-noise of its internal clock. Thus, the ground-plane noise in the endpoint is the combination of the ground-plane noise from the switch's clock (the external clock) *and* from the phase-noise of the internal clock. The external clock phase-noise is thus overlaid *on top* of the noise from the internal clock.

This ground-plane noise subsequently induces jitter on what is producing an output (USB, I^2S etc), (see the first section); thus what is coming out has jitter that is a combination of upstream clocks and the internal clock. If the endpoint is a separate box from the DAC, the data signals carry this overlaid clock signature into the DAC, where it creates ground-plane noise which in turn creates jitter on the clock going to the DAC chip—and that affects the audio signal.

What is important to understand is that this causes a modulation of all audio signals. An example: Let's say the phase-noise is that 10Hz we talked about earlier; you do NOT get a 10Hz noise on the audio output, it changes every signal coming out of the DAC. So if the DAC is outputting 1KHz, you now have 1000Hz, plus 1010Hz, plus 990Hz. Everything coming out of the DAC is "spread out" a bit by this jitter on the clock.

What this clock phase-noise overlay looks like depends upon the clocks and circuits involved. Most "residential" network devices have very cheap oscillators with very high phase-noise, so effects from this in the combined clock can be very strong. Therefore, if your endpoint has a very good local clock, the overlay from upstream can be significantly greater than the local clock.

2) Leakage current:

Leakage current is a property of power supplies. In all power supplies a small amount of current will flow from the AC line, through the power supply, to the DC output. That current will pass to whatever device is being powered, and will always flow back through some path to the AC line, forming a "leakage loop."

Such leakage will frequently flow into the ground-plane of a device, generating a small voltage on the ground-plane. This leakage current is primarily at line frequency—50Hz or 60Hz, depending on the specific country's AC mains standard—though there can be low-level components of it extending up into the megahertz range. Note that this is NOT DC, it is primarily line frequency AC, and that is very important to remember when thinking about leakage loops.

Leakage current can flow through power cables, audio interconnects, digital cables, including Ethernet and USB cables. The leakage will flow from a power supply, through a cable to another device or component, then through that device or component's respective power supply *back* to the AC mains. Alternatively it can flow through a cable to another box and then through a safety ground (the "third pin" of an AC plug) and return back to the AC mains.

We have found there are two types of leakage current, traditional low-source-impedance leakage, which is found in all power supplies, and high-source-impedance leakage, which only occurs in switching-mode

supplies. This high-source-impedance leakage is hard to block and can take surprising paths to get where it wants to go.

These two different forms of leakage currents have a deleterious effect on Ethernet in audio systems. All copper Ethernet connections have transformers in each device port, and these transformers will block DC and low-source-impedance leakage. However, they DO NOT block high-source-impedance AC leakage. This is particularly important for digital networks used for audio.

Most all network systems used with audio use switching supplies (SMPS) to power the digital devices, routers, switches, computers, etc. The high-source-impedance leakage current from the SMPS will travel through the network equipment, the Ethernet cables, the end-points, and into the DAC where they will create noise on the ground-plane, and thus jitter in the DAC circuitry.

Leakage currents can also cause clock modulation in intermediate devices such as “streamers.” The clock issues were discussed in the prior section.

How does the UpTone Audio EtherREGEN address the issues discussed above?

The EtherREGEN’s primary purpose is to radically reduce the two sources of ground-plane noise—leakage current and clock phase-noise—to keep them from getting into your DAC. This is done by preventing leakage (both high- and low-source-impedance types) from passing through, and by preventing data signals from creating phase-noise induced current pulses on the ground-plane.

In the EtherREGEN, leakage is prevented from crossing its ‘A’>’B’ side moat by using specially selected high-speed digital isolators. These block the leakage, but do not block the phase-noise component. Let’s look at common digital isolators to see why. The jitter on the input signal passes through to the output (see prior discussion). Isolators have a delay from input to output; so if the input is a little late the output is also a little late. The jitter passes right on through. Common isolators also produce those infamous pulses on the ground-plane, and it is pretty strong due to how the isolators work. Thus not only does ground-plane noise occur from whatever the isolator is driving, it also occurs from the isolator itself.

It turns out there is a conceptually simple way to fix this. Use fully differential circuitry. This has two wires coming in and two wires going out. The signals are differential; one goes up as the other goes down. Thus a negative and a positive pulse on the ground-plane are created at the same time, and they cancel each other out.

The EtherREGEN utilizes differential, high-speed, very low jitter isolators followed by fully differential ultra-low jitter flip-flops that reclock the data with an ultra-low jitter local clock. Since both of these are fully differential there is no noise on the EtherREGEN’s ground-plane induced by upstream clocks. Additionally, all the clocking in the EtherREGEN is run with differential lines.

It may sound simple, but in fact it was quite challenging. Fully differential low-jitter isolators and flip-flops are extremely rare, and they take multiple voltages and use lots of power. But we did eventually manage to find the right parts and make them work. There are many other unique technical features incorporated into the EtherREGEN. You can read about those on our web site.

Answers to interesting questions:

While previewing a draft of this paper to a few individuals, some questions came back. The answers to these are quite germane to the topic and to our engineering, so we present them below.

Q. Some music renderer endpoints have software or hardware buffers, as do the USB or Ethernet interfaces of some DACs. Do the effects of upstream phase-noise and leakage still impact these?

A. Buffers by themselves do not block phase-noise overlay. As long as there is input data (such as in most USB endpoint buffers) the ground-plane noise from the data still enters the DAC's ground-plane whether there is a buffer there or not. In addition, this phase-noise overlay also occurs inside the buffer itself, plus the input clocks on both the input and output side are subject to the threshold affects of noise on the ground-plane AND on the ground wires inside the buffer chip. So small buffers with data going in and out don't really make much of a difference, and if not done just right can actually make things worse.

USB systems usually have a very small buffer, and since there is always data coming in while data is going out, phase-noise overlay from incoming data is still a factor. Even if the buffer *after* the USB receiver is large, there are always a fair amount of USB packets still on the bus. The only way that goes away is to completely close down the USB connection.

Q. What about systems in which a very large buffer holds a whole song or whole album?

A. A very large buffer where the input completely shuts down while all music is playing can eliminate the phase-noise overlay of upstream sources. However, leakage current is still there. As long as the cable is still plugged in leakage current is still flowing through the cable towards the DAC. So just not sending any packets does not make any difference—the cable has to be unplugged to stop it. Even if there are zero packets from the control system during playback you still have leakage coming from the switch and other upstream network components.

Q. How can these leakage currents be blocked? Don't standard Ethernet ports do this? And what about other types of isolators?

A. For leakage current to be stopped, a true isolator, properly implemented, is required. While you may see discussion of "galvanic isolators," by definition galvanic isolation means only that DC is blocked. Galvanic isolators—be they transformers or any form of capacitive or optical digital logic isolator—still pass AC (alternating current). The transformers in Ethernet ports *are* galvanic isolators—they block DC—they also block low-source-impedance AC leakage. However, high-source-impedance leakage sails right through them.

Now let's look at the case of a USB receiver followed by a digital isolator on the I^2S lines: Depending on the isolator this may or may not block the leakage from upstream power supplies. It is certainly possible to design this so it blocks even high-source-impedance leakage from upstream, but often these circuits do not. This is highly implementation dependent. And as long as data is still coming over that isolator, the output circuit of the isolator itself is generating ground-plane noise on the downstream ground-plane. Then whatever that data goes to is also generating noise on the ground-plane, even if there is a buffer.

Q. What about fiber-optic interfaces? Don't these block everything?

A. In the case of a pure optical input (zero metal connection), this does block leakage current, but it does not block phase-noise affects. The optical connection is like any other isolator: jitter on the input is transmitted down the fiber and shows up at the receiver. If the receiver reclocks the data with a local clock, you still have the effects of the ground plane-noise from the data causing threshold changes on the reclocking circuit, thus overlaying on top of the local clock.