CHAPTER 2

Interfacing Guitar with Computers

A guitar doesn't just plug into a computer the way it plugs into an amp. The guitar's audio must be converted into data before the computer can use it, then converted back into audio so we can hear it.

The Computerized Signal Chain

The computerized signal chain consists of the following stages:

> **Guitar > Preamp > Analog to Digital Converter** (translates analog in to digital out) **> Computer > Digital to Analog converter** (translates digital in to analog out) **> Monitoring system**

Let's explain each block in general terms.

Guitar:

If you don't know what this is, you should probably be reading a different book! But never forget that the guitar has a profound influence on the ultimate sound, no matter what follows it.

Preamp:

A guitar or bass puts out a delicate signal; the preamp beefs it up before it works its way down the signal chain.

Analog-to-Digital converter:

This is built within a device called an audio interface. The audio interface accepts an analog input, and converts it to a data format the computer can understand. Its input will be audio connectors, and its output will be a digital signal cable or other connection that hooks into the computer. See Appendix B: A Guitarist's Guide to Audio Interfaces.

Computer:

This runs the software that processes the data. Think of the host software, such as Pro Tools, Sonar, Logic, Live, Digital Performer, etc. as a "virtual rack" into which you "plug-in" processors—except that these are virtual processors. Amp modeling software processes the guitar sound the same way that amps, cabinets, and effects would.

Digital-to-Analog converter:

This converts the processed data back to analog so we can feed it into a monitoring system and hear it. It's often built into the same audio interface that sends the signal into the computer. Note: Sometimes the audio interface is referred to as "I/O" because it provides the input to the computer and takes the computer's output.

Monitoring system:

This could be a headphone jack in your audio interface, an amplifier with a set of speakers, a PA system, or anything that lets us hear the results of what the computer does.

The Guitar's Special Requirements

Many computers include sound cards, or an onboard audio system. As a result, the computer has audio input and output jacks (typically line and mic in, and line and headphones or speaker outs), and handles all A/D and D/A conversion inside the computer. While onboard sound cards are acceptable for consumer applications, like playing games, they don't give studio quality. Furthermore, the sound card's typical audio inputs are not designed to be compatible with guitar (this can even be a problem with pro gear).

Pro musicians use high-quality audio interfaces that perform the A/D and D/A conversion, and communicate with the computer via a digital signal. These interfaces will have various inputs, though not all are suited for guitar; a guitar's signal is much higher level than a mic's, yet not as strong as a line level signal.

There's also the potential for an impedance mismatch. A standard guitar with stock pickups has a relatively high output impedance, which tends to inhibit an efficient signal transfer to the next in a chain of electronic devices. This transfer is more efficient if the guitar feeds a high impedance input (e.g., more than 220k Ohms) to avoid being loaded down; it's the electronic equivalent of having less "friction." Any significant loading leads to reduced high frequency response, and lower levels. Single coil pickups tend to be more sensitive to loading than humbuckers. Note that active pickups (ones that include preamps) have low impedances and do not need to feed a high impedance input. However, the vast majority of guitars use standard, passive pickups.

There are several ways to avoid loading:

The Studio Guitar Interface conditions your guitar signal so that subsequent circuitry doesn't degrade its tone.

- Use a "buffer" designed specifically for guitar, like the Studio Guitar Interface available from Waves. Designed in conjunction with Paul Reed Smith, it's designed to provide the absolute minimum coloration to your guitar and match its impedance to any subsequent interface.
- Use an active direct box, such as those made by Radial Engineering and others.
- Use an audio interface with an "instrument" input (e.g., from Edirol, MOTU, PreSonus, Yamaha, etc.). These inputs have a high impedance and are intended for guitar or bass.

About Latency (Computer-Based Delays)

Guitar players expect an instant response: Hit a string, hear a sound. That's why it's important to minimize latency—the delay your computer introduces between the time you hit a note on your guitar, and when you hear it come out of the speakers.

Latency occurs in the conversion process from analog to digital and digital to analog, as well as in the computer itself—even the most powerful processor can only do so many millions of calculations per second, and sometimes can't keep up. As a result, the computer sticks some of the incoming audio in a buffer, which is like a savings account for your guitar signal: When the computer is so busy elsewhere that it can't deal with audio, it makes a "withdrawal" from the buffer instead. The larger the buffer, the less likely the computer will run out of audio data when it needs it. But a larger buffer also means that the guitar signal is being diverted for a longer period of time before hitting the computer, which increases latency.

As you want the best possible "feel" when playing guitar, let's investigate how to obtain the lowest possible latency.

Minimizing Latency

The first step in minimizing delay is the most expensive one: Upgrading your processor. With today's dual core processors, and a quality audio interface, it's possible to obtain latencies of well under 5ms—essentially negligible.

The second step involves drivers, little pieces of code that provide communications between your computer and audio interface. They are the data gatekeepers, and how efficiently they do their task greatly affects latency.

Steinberg devised the first universal low-latency protocol for audio interfaces, based on ASIO (Advanced Streaming Input Output) drivers. These tie in closely with the computer's processor, bypassing various layers of both Mac and Windows operating systems. Since then, Microsoft has introduced the WDM protocol to replace their far slower DirectSound and MME protocols, and starting with OS X, Apple gave us Core Audio. Either of these protocols can perform as well as ASIO. However, ASIO remains extremely popular, particularly with Windows.

Drivers are very important, and are frequently updated to improve performance, or insure compatibility with changes in operating systems. It's always a good idea to check a manufacturer's web site for updates, even if you bought a product the day it hit the stores.

Digidesign has its own audio engine, DAE (Digidesign Audio Engine). Pro Tools works only with Digidesign hardware, using their audio engine and driver protocol. However, a few years ago Digidesign "opened up" their audio interfaces so they could be used by other programs. For example, if you're running Sonar, you can choose to have it communicate with a Digidesign audio interface via ASIO.

In any event, some latency is unavoidable. But to put matters in perspective, 3ms of delay is the same as moving 1 meter further away from your speakers. I think most guitarists would agree that latencies below 5ms are fine and between 5 to 10ms are acceptable, but anything much over 10ms is annoying.

However, note that even if your computer can do low latencies in theory, in practice lower latencies stress out your computer more. So, most audio interfaces give you a choice of latency settings, allowing a tradeoff between lowest latency and computer performance. If all your computer has to do is run GTR in standalone mode, then you can probably adjust your system for really low latency. But if you're running a complex digital audio recording program and playing back lots of tracks, you may need to set the latency higher. The symptoms of overloading your system include pops and crackles in the audio, unintended distortion, and sometimes, a program freeze where it simply won't respond.