computers are bad

You are receiving this facsimile because you signed up for fax delivery of this newsletter. To stop delivery, contact Computers Are Bad by email or fax.

https://computer.rip - me@computer.rip - fax: +1 (505) 926-5492

2022-02-19 PCM

I started writing a post about media container formats, and then I got severely sidetracked by explaining how MPEG elementary streams aren't in a container but still have most of the features of containers and had a hard time getting back to topic until I made the decision that I ought to start down the media rabbit hole with something more basic. So let's talk about an ostensibly basic audio format, PCM.

PCM stands for Pulse Code Modulation and, fundamentally, it is a basic technique for digitization of analog data. PCM is so obvious that explaining it is almost a bit silly, but here goes: given an analog signal, at regular intervals the amplitude of the signal is measured and quantized to the nearest representable number (in other words, rounded). The resulting "PCM signal" is this sequence of numbers. If you remember your Nyquist and Shannon from college data communications, you might realize that the most important consideration in this process is that the sampling frequency must be twice the highest frequency component in the signal to be digitized.

In the telephone network, for example, PCM encoding is performed at 8kHz. This might seem surprisingly low, but speech frequencies trail off above 3kHz and so the up-to-4kHz represented by 8kHz PCM is perfectly sufficient for intelligible speech. It is not particularly friendly to music, though, which is part of why hold music is the way it is. For this reason, in music and general digital audio a sampling rate of 44.1kHz is conventional due to having been selected for CDs. Audible frequencies are often defined as being "up to 20kHz" although few people can actually hear anything that high (my own hearing trails off at 14kHz, attributable to a combination of age and adolescent exposure to nu metal). This implies a sampling rate of 40kHz; the reason that CDs use 44.1kHz is essentially that they wanted to go higher for comfort and 44.1kHz was the highest they could easily go on the equipment they had at the time. In other words, there's no particular reason, but it's an enduring standard.

Another important consideration in PCM encoding is the number of discrete values that samples can possibly take. This is commonly expressed as the number of bits available to represent each sample and called "bit depth." For example, a bit depth of eight allows each sample to have one of 255 values that we might label -127 through 128. The bit depth is important because it limits the dynamic range of the signal. Dynamic range, put simply, is the greatest possible variation in amplitude, or the greatest possible variation between quiet and loud. Handling large dynamic ranges can be surprisingly difficult in both analog and digital systems, since both electronics and algorithms struggle to handle values that span multiple orders of magnitude.

In PCM encoding, bit depth has a huge impact on the resulting bitrate. 16-bit audio, as used on CDs, is capable of a significantly higher dynamic range than 8-bit audio at the cost of doubling the bitrate. Dynamic range is important in music, but is also surprisingly important in speech, and a bit depth of 8 is actually insufficient to reproduce speech that will be easy to understand.

And yet, due to technical constraints, 8kHz and 8-bit samples were selected for telephone calls. So how is speech acceptably carried over 8-bit PCM?

We need to talk a bit about the topics of compression and companding. There can be some confusion here because “compression” is commonly used in computing to refer to methods that reduce the bitrate of data. In audio engineering, though, compression refers to techniques that reduce the dynamic range of audio, by making quieter sounds louder and louder sounds quieter until they tend to converge at a fixed volume. Like some other writers, I will use “dynamic compression” when referring to the audio technique to avoid confusion. For both practical and aesthetic reasons (not to mention, arguably, stupid reasons), some degree of dynamic compression is applied to most types of audio that we listen to.

Companding, a portmanteau of compressing and expanding, is a method used to pack a wide dynamic range signal into a channel with a smaller dynamic range. As the name suggests, companding basically consists of compressing the signal, transmitting it, and then expanding it. How can the signal be expanded, though, given that dynamic range was lost when it was compressed? The trick is that both sides of a compander are non-linear, compressing loud sounds more than quiet sounds. This works well, because in practice many types of audio show a non-linear distribution of amplitudes. In the case of speech, for example, significantly more detail is found at low volume levels, and yet occasional peaks must be preserved for good intelligibility.

In practice, companding is so commonly used with PCM that the compander is often considered part of the PCM coding. When I have described PCM thus far, I have been describing linear PCM or LPCM. LPCM matches each sample against a set of evenly distributed discrete values. Many actual PCM systems use some form of non-linear PCM in which the possible sample values are distributed logarithmically. This makes companding part of PCM itself, as the encoder effectively compresses and decoder effectively expands. One way to illustrate this is to consider what would happen if you digitized audio using a non-linear PCM encoder and then played it back using a linear PCM decoder: It would sound compressed, with the quieter components moved into a higher-valued, or louder, range.

Companding does result in a loss of fidelity, but it’s one that is not very noticeable for speech (or even for music in many cases) and it results in a significant savings in bit depth. Companding is ubiquitous in speech coding.

One of the weird things you’ll run into with PCM is the difference between ţ-law PCM and A-law PCM. In the world of telephony, a telephone call is usually encoded as uncompressed 8kHz, 8-bit PCM, resulting in the 64kbps bitrate that has become the basic unit of bandwidth in telecom systems. Given the simplicity of uncompressed PCM, it can be surprising that many telephony systems like VoIP software will expect you to choose from two different “versions” of PCM. The secret of telephony PCM is that companding is viewed as part of the PCM codec, and for largely historic reasons there are two common algorithms in use. The actual difference is the function or curve used for companding, or in other words, the exact nature of the non-linearity. In the US and Japan (owing to post-WWII history Japan’s phone system is very similar to that of the US), the curve called ţ-law is in common use. In Europe and most other parts of the world, a somewhat different curve is used, called A-law. In practice the difference between the two is not particularly significant, and it’s difficult to call one better than the other since both just make slightly different trade offs of dynamic range for quantization error (A-law is the option with greater dynamic range and greater possible distortion).

Companding is rarely applied in music and general multimedia applications. One way to look at this is to understand the specializations of different audio codecs: ţ-law PCM and A-law PCM are both simple examples of what are called speech codecs, Speex and Opus being more complex examples that use lossy compression techniques for further bitrate reduction (or better fidelity at 64kbps). Speech codecs are specialized for the purpose of speech and so make assumptions that are true of speech including a narrow frequency range and certain temporal characteristics. Music fed through speech codecs tends to become absolutely unlistenable, particularly for lossy speech codecs, which hold music on GSM cellphones painfully illustrates.

In multimedia audio systems, we instead have to use general-purpose audio codecs, most of which were designed around music. Companding is effectively a speech coding technique and is left out of these audio systems. PCM is still widely used, but in general audio PCM is assumed to imply linear PCM.

As previously mentioned, the most common convention for PCM audio is 44.1kHz at 16 bits. This was the format used by CDs, which effectively introduced digital audio to the consumer market. In the professional market, where digital audio has a longer history, 48kHz is also in common use... however, you might be able to tell just by mathematical smell that conversion from 48kHz to 44.1kHz is prone to distortion problems due to the inconveniently large common multiple of the two sample rates. An increasingly commonly used sample rate in consumer audio is 96kHz, and “high resolution audio” usually refers to 96kHz and 24 bit depth.

There is some debate over whether or not 96kHz sampling is actually a good idea. Remembering our Nyquist-Shannon, note that all of the extra fidelity we get from the switch from 44.1kHz to 96kHz sampling is outside of the range detectable by even the best human ears. In practice the bigger advantage of 96kHz is probably that it is an even multiple of the 48kHz often used by professional equipment and thus eliminates effects from sample rate conversion. On the other hand, there is some reason to believe that the practicalities of real audio reproduction systems (namely the physical characteristics of speakers, which are designed for reproduction of audible frequencies) causes the high frequency components preserved by 96kHz sampling to turn into distortion at lower, audible frequencies... with the counterintuitive result that 96kHz sampling may actually reduce subjective audio quality, when reproduced through real amplifiers and speakers. In any case, the change to 24-bit samples is certainly useful as it provides greater dynamic range. Unfortunately, much like “HDR” video (which is the same concept, a greater sample depth for greater dynamic range), most real audio is 16-bit and so playback through a 24-bit audio chain requires scaling that doesn’t typically produce distortion but can reveal irritating bugs in software and equipment. Fortunately the issue of subjective gamma, which makes scaling of non-HDR video to HDR display devices surprisingly complex, is far less significant in the case of audio.

PCM audio, at whatever bit rate and bit depth, is not so often seen in the form of files because of its size. That said, the “WAV” file format is a simple linear PCM encoding stored in a somewhat more complicated container. PCM is far more often used as a transport between devices or logical components of a system. For example, if you use a USB audio device, the computer is sending a PCM stream to the device. Unfortunately Bluetooth does not afford sufficient bandwidth for multimedia-quality PCM, so our now ubiquitous Bluetooth audio devices must use some form of compression.

A now less common but clearer example of PCM transport is found in the form of S/PDIF, a common consumer digital audio transport that can carry two 44.1 or 48kHz 16-bit PCM channels over a coaxial or fiber-optic cable.

You might wonder how this relates to the most common consumer digital audio transport today, HDMI. HDMI is one of a confusing flurry of new video standards that were developed as a replacement for the analog VGA, but HDMI originated more from the consumer A/V part of the market (the usual Japanese suspects, mostly) and so is more associated with televisions than the (computer industry backed) DisplayPort standard. A full treatment of HDMI's many features and misfeatures would be a post of its own, but it's worth mentioning the forward audio channel.

HDMI carries the forward (main, not return) audio channel by interleaving it with the digital video signal during the "vertical blanking interval," a concept that comes from the mechanical operation of CRT displays but has remained a useful way to take advantage of excess bandwidth in a video channel. The term vertical blanking is now somewhat archaic but the basic idea is that transmitting a frame takes less time than the frame is displayed for, and so the unoccupied time between transmitting each frame can be used to transmit other data. The HDMI spec allows for up to 8 channels of 24-bit PCM, at up to 192kHz sampling rate--although devices are only required to support 2 channels for stereo.

Despite the capability, 8-channel (usually actually "7.1" channel in the A/V parlance) audio is not commonly seen on HDMI connections. Films and television shows more often distribute multi-channel audio in the form of a compressed format designed for use on S/PDIF, most often Dolby Digital and DTS (Xperi). In practice the HDMI audio channel can move basically any format so long as the devices on the ends support it. This can lead to some complexity in practice, for example when playing a blu-ray disc with 7.1 channel DTS audio from a general-purpose operating system that usually outputs PCM stereo. High-end HDMI devices such as stereo receivers have to support automatic detection of a range of audio formats, while media devices have to be able to output various formats and often switch between them during operation.

On HDMI, the practicalities of inserting audio in the vertical blanking interval requires that the audio data be packetized, or split up into chunks so that it can be divided into the VBI and then reassembled into a continuous stream on the receiving device. This concept of packetized audio and/or video data is actually extremely common in the world of media formats, as packetization is an easy way to achieve flexible muxing of multiple independent streams. And that promise, that we are going to talk about packets, seems like a good place to leave off for now. Packets are my favorite things!

Later on computer.rip: MPEG. Not much about the compression, but a lot about the physical representations of MPEG media, such as elementary streams, transport streams, and containers. These are increasingly important topics as streaming media becomes a really common software application... plus it's all pretty interesting and helps to explain the real behavior of terrible Hulu TV apps.

A brief P.S.: If you were wondering, there is no good reason that PCM is called PCM. The explanation seems to just be that it was developed alongside PWM and PPM, so the name PCM provided a pleasing symmetry. It's hard to actually make the term make a lot of sense, though, beyond that "code" was often used in the telephone industry to refer to numeric digital channels.