Digital Audio Systems

Fixed Point Or Floating Point?

Digital audio systems use a form of binary coding to represent the audio data. The traditional system uses a fixed-point representation (pulse code modulation or PCM) that employs identical resolution steps to quantize the data throughout the full dynamic range. Increasing the resolution requires an increase in the number of bits. Sixteen or 24-bit PCM is used by CDs, digital tape recorders, many Flash RAM recorders, and previously by digital audio workstations (DAWs) like Pro Tools. Alternatively, the data can be encoded in a floating-point representation. Floating-point numbers consist of a mantissa (data value) and an exponent (scale factor). By using scaling, an encoded signal is able to use the entire accuracy of the measurement whether the signal is small or large. In essence, the exponent moves the decimal point within the mantissa. Early computers used for digital audio were fixed-point systems. Modern personal computers use floating point and digital audio has followed. In order to appreciate the significance of the difference between fixed and floating point math, we need to consider the difference between accuracy and precision.

Precision is the closeness of two measured numbers, in our case audio samples, while accuracy is the proximity of the measurement to the actual value. For example, an A/D converter is precise if it gets the same value each time but it is accurate only if that value is the exact value of the sampled analog signal at the time of the sample. (A golfer is precise if he hits the same spot on the green each time: he is accurate if that spot is the hole.) Floating point covers a wide dynamic range of measurements, encoding very large and small numbers by changing the exponent. But it still only accurate to the 24 bits of the mantissa. Its advantage for audio can be appreciated when considering very small numbers, which are important in audio as a signal fades to silence. By scaling the value using the exponent, all 24 bits of the mantissa can be applied to the very small values. In fixed point, the same small values are represented by only the lowest significant bits and dither must be added to reduce the error of the measurement. While floating point numbers maintain better precision for small numbers, the changing exponent makes applying dither problematic. Dither is needed, for instance, when a floating point value must be converted to a fixed point value as would be needed to feed a D/A converter or create an audio CD or DVD.

Comparison of fixed- and floating-point performance can be confusing. For fixed-point systems, the signal-to-noise ratio (measured with signal present) and the dynamic range (measured with no signal present) are closely related. Floating-point systems perform differently because the A/D converter alone, which gives the mantissa measurement, determines the signal-to-noise ratio while the exponent also affects the dynamic range. Floating-point systems generate changing quantization error as a function of the exponent change while fixed-point systems produce fixed quantization error that is significant only at the lowest amplitudes. Since 24 bits are more than enough to encode and reproduce audio signals, it is when sample values are processed in DSP that the advantages of floating point systems become apparent.

Once we have a digital representation of the audio signal, it may be processed mathematically like any other computer data. This leads to the ability to simulate the behavior of many analog signal-processing devices in the digital domain. If we know the behavior of the simple electronic devices, we can model them using software to create virtual devices that can process the digital data like the analog devices do with analog audio signals. By using longer word lengths (double or triple precision), calculations can be accomplished without generating audible errors and the final data may then be returned to the original bit depth through dithering. The optimization of this conversion allows a transparent processing that closely models the analog device but still returns the original amount of data after the process.

Current Analog-to-Digital Techniques

The older method of converting analog signals to digital, the successive-approximation method, sampled at the desired rate and required a low-pass filter to remove any signal components above half the sample rate. These analog low-pass filters required many series stages to give a sharp enough roll-off to eliminate the unwanted frequencies while preserving the entire bandwidth below the cut-off frequency. The heavy filtering noticeably altered the sound of the higher audio frequencies.

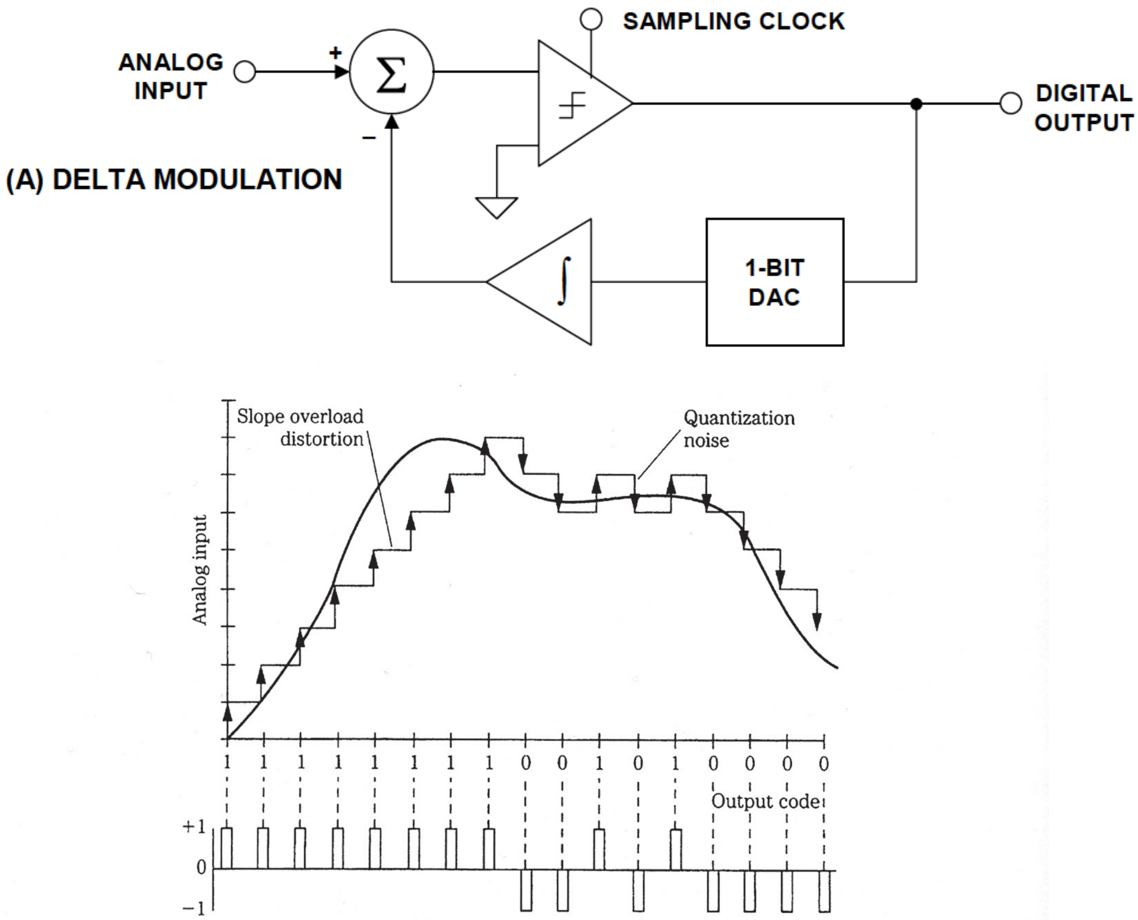

Figure 1: Delta modulation

The more modern solution is to sample at much higher frequencies, allowing gentler analog filtering, and use a digital filter to ultimately remove the unwanted signal components. The A/D converter is run at much higher sample rates than the desired final rate and then filtered digitally. Digital filters can be generated that do a much better job than the analog filters. The quantizer may use as little as one bit but at a very high rate. These bits are then summed to produce multi-bit words in a technique know as sigma-delta modulation. This technique is called oversampling. The technique uses a decimation filter to produce the final sample rate and bandwidth, effectively resampling the data stream.

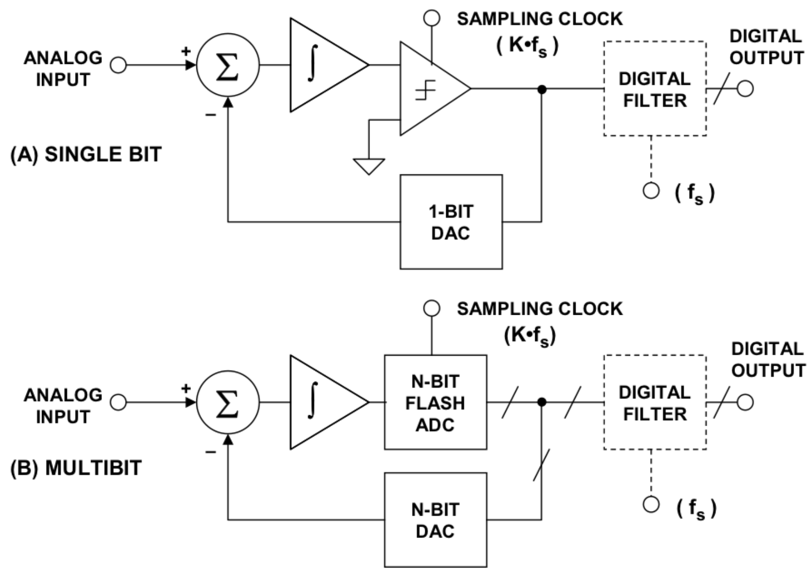

Figure 2: Sigma-delta modulation

Most current audio A/D converters use the sigma-delta approach, which simplifies the quantizer by using a single bit (or low-bit) conversion running at very high sample rate and converting the generated data stream to multi-bit by decimation filtering. The modulator can use a one-bit quantizer or a multi-bit quantizer that uses a small number of bits to maintain linearity, four bits for example. This eliminates several potential non-linearities inherent in the multi-bit converter and allows dithering that is not available for a one-bit quantizer.

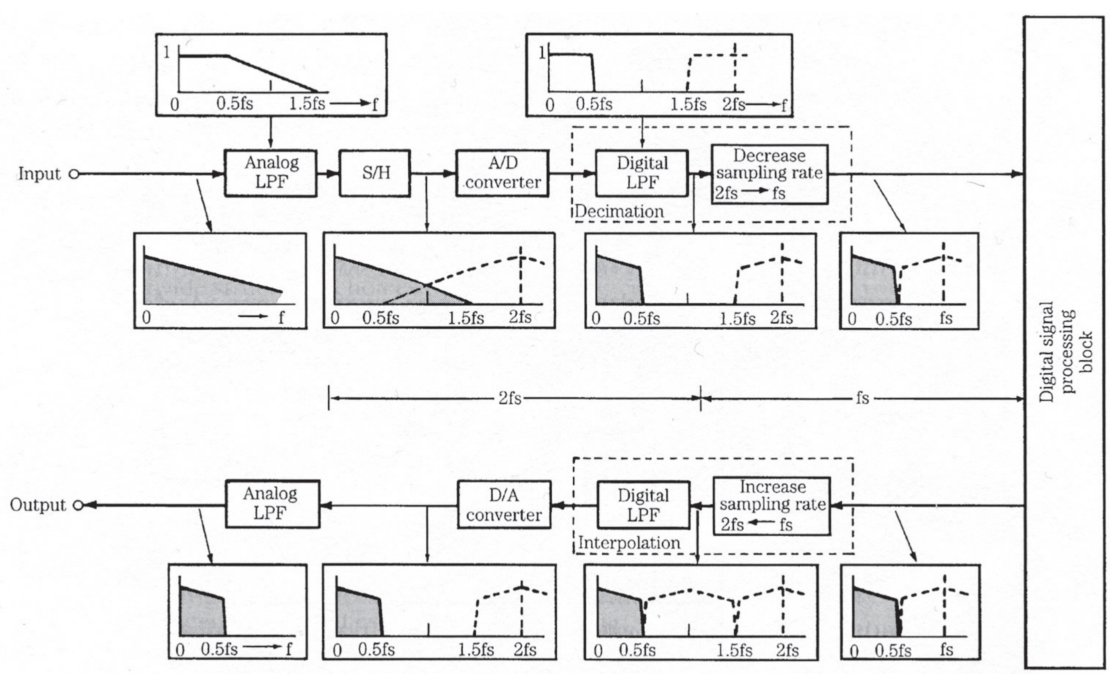

Figure 3: Overall oversampling A/D and D/A system

Error Detection and Correction

Analog audio systems have a distinct advantage over digital systems when it comes to disruptions of the signal flow. No matter how long an analog disruption persists, the signal always comes back in synchrony. While we may notice a drop-out that reduces the fidelity of the sound, we do not lose our place in the signal. Since digital audio transmits words made up of sequential bits, a disruption of the signal can cause the meaning of the bits received to become ambiguous and the signal then cannot be properly reproduced. Simply transmitting the raw digital stream from the A/D conversion leaves us susceptible to complete loss of the information should a drop-out or other disruption occur.

Inside a computer, the stored data is carefully preserved. A single bit change cannot be tolerated as it could change an instruction from an add to a multiply, for example. For this reason, computers have systems to guarantee the fidelity of stored information that satisfactorily protect all data. But when digital audio is outside the confines of the computer, these guarantees do not apply. If digital audio is transmitted from one place to another, it becomes susceptible to alteration by external interference coupling or unreliable connections. For this reason, digital audio systems have developed their own methods of error detection and correction.

There are always inherent problems associated with the storage and retrieval of audio signals. In the case of analog recorders, the main limitations are due to the physical constraints of the media: noise, distortion, and dropouts. While digital magnetic and optical recording systems do not suffer directly from noise or distortion, they are susceptible to dropouts. When a dropout occurs, data is lost. The longer it lasts, the more data is lost. While analog systems also suffer from dropouts, it is unusual for the signal to be entirely corrupted: it generally loses amplitude and possibly some of the high frequency content but when it returns, it is still the correct signal. When a dropout disrupts a digital data stream, there is no way to tell what is missing and time synchrony may be lost entirely. Therefore, digital audio systems must either be able to ensure complete data integrity (as is the case of solid-state memories like DRAM computer memory) or provide for the detection and correction of any data errors that are likely to occur.

The simplest way of detecting errors is to transmit the data several times and compare the results, but this adds greatly to the required storage space and transmission bandwidth. Another approach is to simply add up the sum of a certain number of words and store the sum with the data. If the replayed data does not add up to the so-called check sum, the data has an error. This system may be able to detect errors, but it cannot fully reconstruct the original data. Another way to minimize the effect of data loss is to spread the bits of each word out between the bits of previous and subsequent words: the process known as interleaving. This reduces the likelihood that all the bits of any word are lost, making the job of reconstructing the data a bit easier. To further increase the ability to determine what the original data were after a dropout, more data may be added to the signal in a way in which it can be used to reconstruct the original data even if large amounts of data are lost. These methods use sophisticated mathematical processes to allow data reconstruction and are known collectively as error detection and error correction.

Interleaving

Interleaving is the process of distributing data from data words over wide areas of the tape. This is accomplished by mixing bits from several words in a fixed way along the tape. When a dropout occurs, it will damage the data from several adjacent bits, but probably only one or two bits from each word will be altered, instead of all of the bits from one or two words. This allows for better reconstruction of the original data. A further development of this scheme is cross-interleaving, which uses two distinct codes and separates them by a time delay, further insuring that the error will be correctable.

Parity

A bit is added to a string of bits that indicates if the sum of the bits is odd or even. This is a rather crude method of indicating an error has occurred, since there are combinations of random changes that will still provide the same parity.

Checksum

A more useful type of error detection is to sum the data over some period and store the sum. This gives some ability to recover the original data if the checksum fails to match on playback.

CRCC

This stands for cyclic redundancy check code: a form of checksum. Data is gathered into blocks, divided by an arbitrary constant and the remainder is appended to the data. On playback, the data is again divided by the same constant: a remainder of 0 indicates valid data. This scheme works best for burst errors (errors involving short periods of data disruption).

Reed-Solomon Code

This scheme uses Galois fields (number sets which when raised to powers or arithmetically combined always generate another member of the field.) This allows regeneration of data since only specific values can be contained in the original data.

CIRC

Cross-interleaved Reed-Solomon coding: used in CD coding. Uses two Reed-Solomon codes with an inter-leave between.

Coding and Modulation Schemes for Digital Audio

Modulation

Modulation refers to the combination of two or more signals into a composite that conveys information from both. Examples with which we are familiar include amplitude (AM) and frequency (FM) modulation used in radio communications. If we were to record raw audio data directly to tape, we would need an enormous bandwidth, since each bit would need to be coded as a distinct symbol with an on and off, each requiring a polarity reversal in the magnetic medium. We also need to transfer the clock information to keep everything synchronized. A method of reducing the number of distinct transitions is required, preferably one that also contains information about where in the bit stream a data word begins. Also, consider the case of a stream of consecutive 1's...this would provide a constantly high voltage, and consequent magnetic polarization, which would act like a DC signal and could eventually polarize the magnetic heads, as did the dc bias used in early analog magnetic recorders. Our modulation scheme should guarantee that DC does not appear in the signal.

Pulse code modulation

The basic coding scheme used in digital audio is pulse code modulation: signal amplitudes are measured in 16-

bit A/D converters and sent out as serial data in which a logical one is coded by a positive voltage and a logical zero is coded by a zero (or negative) voltage. This scheme does not directly convey clock information when sequential bits do not change, nor does it allow for error detection, so further modulation is necessary.

Eight-to-ten modulation

Many coding schemes exist, and several are employed in computer disk drives and digital audio systems. The DAT uses eight-to-ten modulation, where 8-bit bytes are encoded as 10-bit symbols uniquely assigned to represent the possible 8-bit bytes. This allows the choice of symbols with an exact balance of 1's and 0's, eliminating DC from the system. It also maintains a relatively low channel bandwidth by eliminating sequences of alternating 1's and 0's. The limited wavelength range generated by this modulation code also allows the signal itself to erase an underlying signal, thus eliminating the need for an erase head.

Eight-to-fourteen modulation

The compact disk uses a different modulation scheme. Instead of 8-10 modulation, the CD uses 8-14 (EFM) modulation. EFM maximizes the number of transitions possible with an arbitrary pit and land length, determined by the wavelength of the laser light used to read the data. (On CDs, ones are coded as edges of pits and zeros as pits and lands.) EFM modulation allows high-density recording by reducing the effective minimum wavelength below the resolving power of the laser.

Digital Audio Transmission

Digital audio signals can be routed between devices much like analog signals, although the actual signal is digital. Two commonly used protocols are used for stereo digital signal transfer: the AES/EBU(AES3) and S/PDIF(IEC 958 type II) systems. The AES/EBU (Audio Engineering Society/European Broadcast Union) system uses a differential (balanced) bit-serial high voltage (3-10 V) data stream that contains audio and other (sub-code) data of a specific form. The S/PDIF (Sony/Philips Digital Interface Format) system uses unbalanced low-voltage (0.5 V) bit-serial code with slightly different sub-code definitions. The AES/EBU protocol uses XLR connectors and is considered a professional audio format, while the S/PDIF system (also known as IEC-958 type II) uses RCA connectors and is intended as a consumer format. Both formats send data at almost 3 Mbits/sec, so special cabling must be used for long distances.

Because these systems are "self-clocking" (that is, they carry sample clock information as well as audio and sub-code data), special care must be taken to prevent jitter, drift in the clocking accuracy, from degrading system performance. Cabling of the proper impedance must be employed to keep internal electrical reflections from occurring in the cabling and confusing the D/A converter. Both the S/PDIF and AES/EBU systems expect properly terminated connections: the impedance of the input must be matched to the system impedance: 75 ohm for S/PDIF and 110 ohm balanced for AES/EBU.

In addition to the stereo digital interface formats, there are several multi-channel formats. With the ADAT came the 8-channel ADAT lightpipe protocol for digital audio interchange. This protocol uses optical fiber to transmit 8 channels in one direction, so that input and output are transmitted separately. The ADAT lightpipe is self-clocking and can transmit up to 24-bit words. Tascam introduced their multi-channel protocol with the DA-88. Called TDIF, this protocol is bidirectional and carries 8 channels and a clock over wires contained in a single cable for both directions. The MADI (AES10) Multi-channel Audio Digital Interface protocol allows up to 56 channels of digital audio to be transmitted on coaxial cable up to 150 feet. MADI is limited to 48 kHz sample rates, although higher rates may be implemented by sharing channels. MADI is unidirectional, requiring two cables plus a clock for bidirectional connection. If optical transmission is used, the MADI system may be used over distances up to 2 km. A multi-channel protocol from Yamaha is the mLan system, using FireWire hardware interfaces to send multi-channel digital audio and/or MIDI data. Up to 100 audio channels may be transmitted over an IEEE-1394 cable. A more recent digital audio protocol is Dante from Audinate, which allows 512 uni-directional audio channels at 44.1/48 kHz sample rates to run over standard Ethernet networks. This approach avoids the proprietary hardware approaches of the past but is still a proprietary software solution. An open-source alternative is AVB (Audio-Video Bridging), which uses the same Ethernet network but with non-proprietary software. It also supports video with tight synchronization.