Test Strategies Confront Streaming Audio Quality Challenges

By Bjorn Hori and Jeff Bier, Berkeley Design Technology, Inc.

When they work well, Internet audio devices seem to be the essence of simplicity: press "play" and hear your music. But lurking under the minimalist user interface is a surprising amount of complexity. From a designer's perspective, this means lots of things can go wrong. Indeed, when *Consumer Reports* magazine tested MP3 players recently, one widely available model exhibited severe audio-quality glitches, reinforcing the adage that if you don't find the bug, your customers will. Ensuring that users have a high quality listening experience requires a well-thought-out testing strategy—one that starts early in the design process.

Thorough testing of streaming audio devices is made particularly challenging by the subjective nature of audio quality, the large amounts of data required for testing, the large number of operating modes of typical audio decompression algorithms, and the need to ensure robust real-time performance with a very low-cost processor.

Audio Quality

Streaming audio compression algorithms like MP3 are "lossy" or "perceptual" compression algorithms. This means that after compression and decompression, the output is not an exact reproduction of the original. The compression algorithm eliminates portions of the signal that are judged imperceptible. The compression algorithm also quantizes the signal, which introduces noise in a carefully controlled manner--noise a listener will not hear because it will be masked by other sounds. Signal elimination and quantization complicate testing, since traditional audio test metrics like signal-to-noise ratio will indicate significant degradation of the signal, but a listener may not hear those differences. Similarly, quantitatively minor errors in the signal (like a single output sample with the wrong value) may be very objectionable to a listener.

Given the difficulties in measuring the audio quality of a lossy compression algorithm, decoders are typically tested with special test vectors and the output is compared to the output of a "reference" decoder. Test vectors and reference decoders are usually supplied by the vendor of a compression algorithm or by a standardization body, such as the ISO/IEC (which handles MP3).

The reference output is usually generated with a decoder that uses floating-point arithmetic, while embedded streaming audio devices typically use fixed-point arithmetic for cost reasons. Ideally, the output of the fixed-point decoder will match that of the floating-point reference decoder, but this depends on the quality of the fixed-point implementation. Creating a high quality fixed-point decoder implementation requires an in-depth understanding of the algorithm's potential dynamic range and of error

propagation with fixed-point math. Lesser quality fixed-point implementations may perform reasonably well with benign audio content, but fail when faced with more challenging content. The supplied test vectors may have been designed with floating-point arithmetic in mind, and may not adequately stress a fixed-point implementation. Thus, to ensure thorough testing of a fixed-point implementation it is essential to create tests that exercise the full potential dynamic range of the algorithm.

High Data I/O Requirements

Streaming audio devices, particularly those that support high-quality audio, consume and produce vast amounts of data. Data arrives from an external source, is quickly processed, then sent out to an output, never to be seen again. Thus testing a streaming audio device requires the ability to inject input streams and capture output streams. And while the end user typically will access output streams in analog form after digital-to-analog conversion, the developer will want to capture the output digitally for testing.

One key question is how to test the compression algorithm implementation prior to the availability of product prototypes. Because large amounts of data are needed for testing, a simulation model of the processor is usually far too slow—real hardware is needed. When testing audio software with a development board, it is critical to ensure that there is a means for streaming large amounts of compressed audio into the processor and large amounts of decompressed audio out of the processor. For developers, this means that carefully choosing a development platform with appropriate I/O capabilities can save time and reduce headaches when the testing phase begins.

Operating Modes

For each compression algorithm used in a streaming audio product, all operating modes must be tested. An operating mode consists of a valid combination of bit-rate, sampling rate, and channel configuration. Typical audio compression algorithms support several choices in each of these categories, and the number of valid combinations is high. For example, MP3 decoders should support at least fourteen bit rates (32, 40, 48, 56, 64, 80, 96, 112, 128, 160, 192, 224, 256, and 320 kbps), three sampling rates (32, 44.1, and 48 KHz), and four channel configurations (single, dual, stereo, and joint-stereo); many support more.

Real-Time: Not Optional

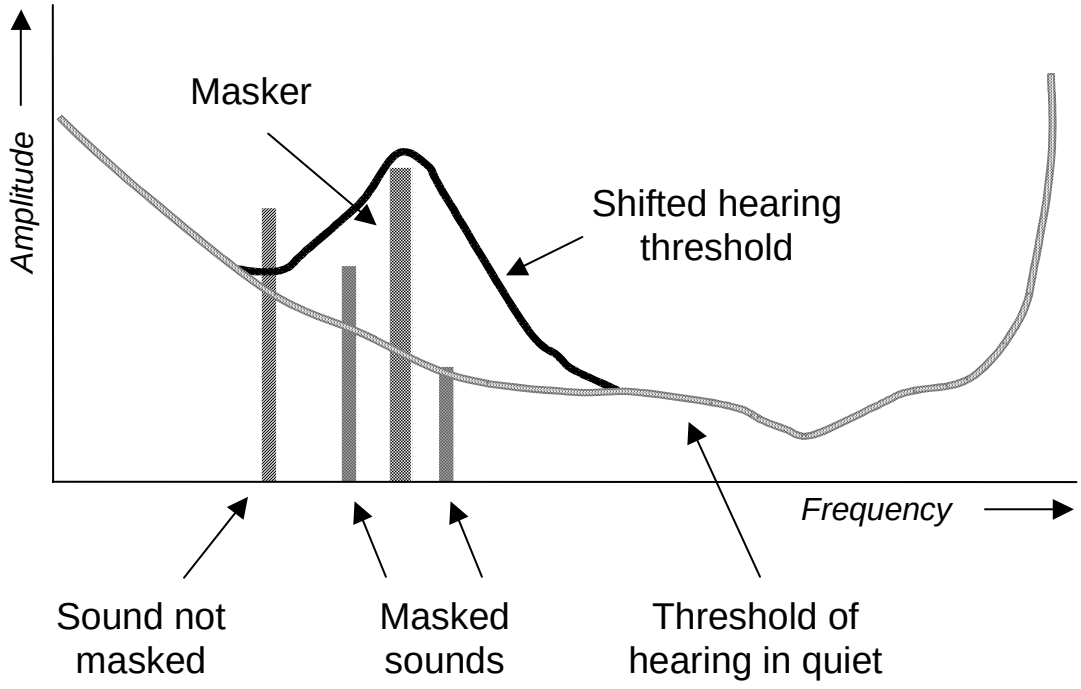
The need for consistent real-time performance is clear; without it the audio output stutters and coughs. Due to severe cost constraints, underpowered processors are frequently pushed into service in streaming audio products. Careful optimizations of the compression algorithm implementation can often pull these underachievers up to real-time performance levels, but the proof is in the testing.

To stress the limits of the system's real-time behavior, developers typically test the most demanding operating mode; e.g., the highest bit rate, at the highest sample rate, and with the greatest number of channels. Although this certainly is a good start, there's no guarantee that this approach actually tests the worst case. This is because compression

algorithms typically have data-dependent execution paths; that is, the path that the processor takes through the software depends heavily on characteristics of the input data. Further complicating matters, in some processors the number of cycles required to execute basic operations such as multiplications is data-dependent. Thus, to successfully test worst-case real-time performance, the most demanding operating mode must be combined with an input stream that ensures that the worst-case execution path is taken and that any operations with data-dependent timing are presented with worst-case inputs.

Don't Skimp on Testing

In the high-stakes game of consumer electronics, having your product fail in the field, or delayed in getting to market, can be a disaster. A key for managing such risks is to recognize that despite the surface simplicity of Internet audio products, they are complex systems that require a rigorous approach to testing—one that begins very early in the design process.



One sound can mask another, making it inaudible. Inaudible sounds are eliminated, while audible sounds are quantized. Quantization reduces the number of bits required to represent the sound, but introduces noise.

Bjorn Hori is a DSP Engineer and Jeff Bier is General Manager at Berkeley Design Technology, Inc. (www.BDTI.com), an independent DSP technology analysis and software development company. BDTI's Web site provides a wealth of free information of interest to designers of Internet audio devices.