An Industry Standard Benchmark Consortium

DENBench™ Version 1.0

Benchmark Name: MP3 Decode

Highlights

- **Benchmarks potential performance of an MP3 player's processor subsystem**
- **Uses five different test files**
- **Integer implementation derived from the MSSG ISO sources**
- **Implements PSNR to check the output quality**

Application

The MP3 Decoder benchmark provides an indication of the potential performance of a microprocessor subsystem running an MP3 player application.

Benchmark Description

The benchmark is an integer implementation of the ISO 13818-3 MPEG-2 Layer 3 decoder with lower sampling frequency extension. Normal sampling frequencies are 32 kHz, 44.1 kHz (typical CD-ROM audio), or 48 KHz. Lower sampling frequencies are 16 KHz, 22.05 KHz, or 24 KHz. We selected a lower sampling to reflect that which is often used in PDAs, mobile phones, and on websites where bandwidth is a concern. The benchmark does not include the standard MPEG optimizations, i.e. neither the 0.9.3 nor the 0.9.5 optimizations are implemented because we selected pure "reference code" as the baseline for this benchmark.

The benchmark includes both Huffman decoding and modified inverse discrete cosine transform (iDCT) routines.

The benchmark simulates the decoding and playback by encapsulating data statically (rather than through file I/O) of the following MP3 encoded files:

- JUPITER.mp3: A faithful rendition of "Jupiter, the Bringer of Jollity" from Gustav Holst's *The Planets*. Encoded at 160 KBps, stereo. Dynamics suggest a full range of signals.
- music128stereo.mp3: A sophisticated set of music and noise samples spanning the full dynamic range. This one is encoded at 128 KBps (constant), stereo, at very high quality, and consists of one minute's worth of playback. This file is about 993KB on disk (based on Windows XP NTFS filesystem). 128 KBps is considered the best MP3 rate for quality in portable players.
- music48_128stereo.mp3: Same set of music as above, but encoded in stereo with a variable bit rate of between 48 KBps and 128 KBps. Very high quality. 855 KB on disk suggests that most of it was encoded at high bit rates by our encoder at the EEMBC Certification Laboratory (ECL), but some at lower bit rates. 48 KBps is often the maximum bit rate used for cellular/mobile phone.
- music64stereo.mp3: Same set of music as above, but encoded at 64 KBps, an ideal compromise on quality vs. size. Constant rate, stereo. 470 KB on disk.
- music48mono.mp3: Same set of music as above, but encoded in monaural (mono), not stereo, at 48 KBps constant. 353 KB on disk.

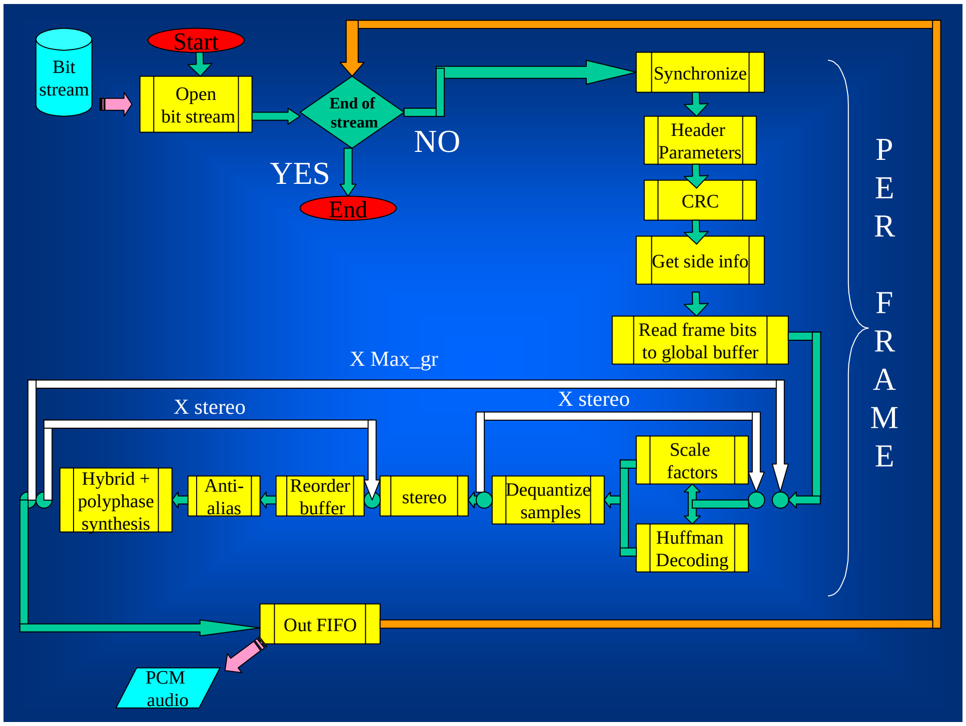
Processing consists of:

1. Reading the selected MP3 file.
2. Reading and interpreting the header information.
3. Read and decode frames of data.
4. Process the data based on the header information.
5. Output music as two channels of 16-bit pulse code modulated data. This data is placed into an AIFF format file which is supported by a wide range of players.
6. A PSNR value is calculated for each PCM frame against a reference AIFF file. These frame scores are aggregated into a single PSNR score for the benchmark.

A single iteration of the benchmark is complete when the end of the input file is reached and no more data is available to be processed.

A way to measure the quality of the output based on peak signal-to-noise ratio (PSNR) code was developed by ECL and implemented in accordance with the Consumer Subcommittee, Technical Advisory Board (TechTAG), and EEMBC Board of Directors. PSNR is a decibel measurement of noise power. PSNR is consistent for the industry and widely used to measure picture quality and audio quality. PSNR is measured outside the benchmark timing loop, on the host side processor (not the embedded target).

The flow diagram is as follows:



An Industry Standard Benchmark Consortium

Analysis of Computing Resources

This is an integer-only implementation of MPEG-1/2 Layer 3 audio and is a benchmark that concentrates mostly on computational processing rather than file I/O. In the following order, synchronization and error checking, Huffman decoding, re-quantization (using inverse discrete cosine transform, iDCT), and reordering are performed.