Implementation of a High-Quality Dolby* Digital Decoder Using MMX™ Technology

James C. Abel and Michael A. Julier
Intel Corp., Chandler, AZ

Abstract

Software decoding of Dolby Digital allows it to become a baseline capability on the PC, with greater flexibility than a hardware approach. Intel's MMX™ technology provides instructions that can significantly speed up the execution of the Dolby Digital decoder, freeing up the processor to perform other tasks such as video decoding and/or audio enhancement.

Intel has worked closely with Dolby Laboratories to define an implementation of Dolby Digital based on MMX technology that has achieved Dolby's certification of quality.

Introduction

The goal of this investigation was to find a minimal CPU implementation that would pass Dolby Laboratories certification for quality (analytical and subjective tests).

Intel's MMX instructions operate on 8, 16, and 32 bits. The smaller the data size, the more operations per instruction are performed. Using 16 bits of accuracy uniformly through a Dolby Digital decoder is insufficient to pass the test suite. The challenge was to obtain both good execution speed and good audio quality. 32-bit floating-point numbers could be used throughout the data path and only use MMX technology for bit manipulation, but this would not be the most processor-efficient method. To this end, we used 16-bit SIMD (Single-Instruction, Multiple Data) operations during much of the decoder, but performed 8 and 32-bit SIMD operations on certain sections.

Dolby Digital Decoder

During our investigation, each functional block of the Dolby Digital Decoder was inspected to determine if it could benefit from MMX technology. The following operations benefit significantly from MMX technology: Bit Stream Parsing, Scaling, TDAC Transform (DCT twiddles, FFT, Windowed-Overlapped-Add), Dithering and Downmixing.

We will now describe the five major operations from the input to the output (Bit Stream Parsing, Coefficient Extraction, TDAC Transform, Dithering, and Downmixing). We will also describe how MMX technology was used to provide a speedup. General precision and performance enhancements will also be discussed.

Bit Stream Parsing

Each audio block contains various pieces of information that tell the decoder how to decode the audio. These are bit fields that are extracted M bits at a time, where M is 0 to 16. MMX technology can be used to perform bit extraction [1], so we can efficiently parse the bit stream. From this information, we obtain the transform coefficients for the synthesis filter bank (TDAC transform).

TDAC Transform

The Time Domain Aliasing Cancellation (TDAC) transform [2] converts the spectral information back to time domain, pulse-code modulated (PCM) samples. The TDAC provides perfect reconstruction (in the absence of quantization or other noise) and is critically sampled.

The TDAC transform is implemented as two DCT twiddle stages with an inverse Fast Fourier Transform (iFFT) in the middle [2]. A block diagram of this implementation of the TDAC transform is shown in Figure 1.

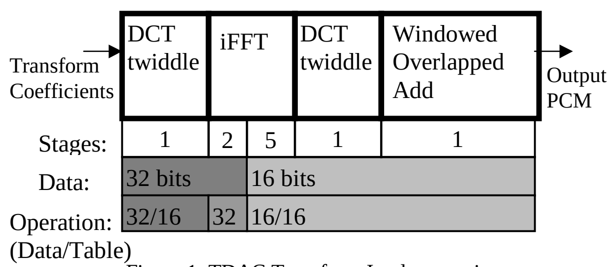

Figure 1. TDAC Transform Implementation

In our implementation, we created coefficient values with 24-bits of accuracy that are stored in 32-bit values. 24 bits of accuracy was chosen to prevent overflow during the intermediate denormalization and scaling processes. This 32-bit number was used in the first three stages of the TDAC transform. After the first two stages of the iFFT, the value was rounded to 16 bits of accuracy. The remainder of the operations were performed using pass-to-pass representations of 16 bits. MMX technology provides multiply accumulations to 32 bits, therefore many intermediate values were 32 bits.

The sine, cosine, and windowing values required in the TDAC transform were implemented via 16-bit lookup tables. Since these values are full-scale, 16 bits was sufficient for our needs. Errors introduced by imprecise coefficients are negligible compared to roundoff errors [3,4]. The technique of 32-bit data and 16-bit lookup tables has been shown to provide high-quality audio decoding [3].

Quantization errors introduced early in the transform process manifest themselves as tones in the output. Tonal noise is highly objectionable [4]. Output noise, if it must be present, should be broad-band or "white" noise. Therefore, the goal was to significantly reduce the peak spectral error. In a mixed-precision implementation, the question is: how far into the TDAC transform do we need to carry 32 bits? In other words, where can we switch to 16 bits? Under subjective listening tests, we decided that performing the first three stages in 32 bits and the remainder in 16 bits reduced the tonal noise to a level of acceptability (see Figure 3). This also resulted in the decoder passing the measurement tests.

Multiplication in the MMX instruction set is 16 bits by 16 bits, yielding a 32-bit result. A 16-bit by 31-bit multiply is also possible in software, at a cost of at least five instructions [5]. Minimizing the number of 16-by-31 bit multiplies was important. It was discovered that the first two stages of the Decimation in Time (DIT) FFT contain only trivial coefficients, i.e., -1 and +1. This allowed these stages to be performed using only add and subtract instructions (no table lookup operations). These 32-bit operations are available in the MMX instruction set. This optimization allowed us to use the more computationally intensive 16/31 bit operations only on the first DCT twiddle stage. The first two stages of the iFFT were performed with 32-bit adds and subtracts, which are efficient in the MMX instruction set.

The Windowed-Overlapped-Add (WOLA) block also fits well into the MMX instruction set. To perform the WOLA, the current and previous output arrays from the last DCT twiddle stage are windowed and then added together [6]. The windowing and addition operations were implemented as two 16-bit by 16-bit multiplies (the windowing) and then added as 32-bit quantities. This is provided by the PMADDWD instruction. The 32-bit results were then rounded to 16 bits for the output.

Mantissa Dithering

Dithering is required in a Dolby Digital decoder. This is implemented as a pseudo-random number generator that is random to 14 bits (the Dolby Digital specification states that the random number generator must be random to 8 bits or greater [7], so we exceed that specification). The calculation is given in Listing 1.

Listing 1. Dither Generation

```
C code:

x(t) = (x(t-1) * 0xbb75) & 0xffff;

MMX Technology Assembly Code:

// dither multiplier is linear congruential
// multiplier ^ 4, 0x4f31, packed 4 times
Quadword DithMultVal = 0x4f314f314f314f31;

// [63:48] = 0xbb75 ^ 4   [47:32] = 0xbb75 ^ 3
// [31:16] = 0xbb75 ^ 2   [15: 0] = 0xbb75 ^ 1
Quadword DithregInit = 0x4f31994d2379bb75;

Initialization:
        ;4 16-bit packed values
        MOVQ    MM0, DithregInit
Generation Loop:
; dither register * dither multiplier to
; get next set of values in dithe register
        PMULLW  MM0, DithMultVal
        ;result is 4 16-bit values
        MOVQ    [result64], MM0
```

Calculating four dither values with a single PMULLW instruction provides a high throughput for this part of the decoder. This instruction multiplies two 16-bit values and provides the lower 16 bits of the result (four of these are performed per instruction).

Downmixing

Dolby Digital can contain up to six ("5.1") audio channels, so downmixing to two channels is often used (LFE is assumed to be discarded). Downmixing is generally an additive process. Scaling (which is discussed below, see "Early Scaling") is also part of downmixing in Dolby Digital. It is used to set relative levels between downmixed channels. Since we perform it up front as part of the denormalization process, downmixing becomes additive. MMX technology provides SIMD addition, which speeds up downmixing.

Precision Enhancements

Precision enhancements such as Rounding and Gain Ranging increased the processing requirement slightly, but resulted in a significant quality improvement and were judged to be worth the additional overhead.

Rounding

It is important to perform rounding when a higher-precision number is being converted to a lower-precision number (e.g., 32 to 16 bits). For example, the PMADDWD instruction (packed multiply-accumulate) multiplies 16-bit numbers, yielding a 32-bit result. If this 32-bit result is to be converted to a 16-bit value, rounding should be used. Rounding can provide a significantly reduced error compared to truncation [8]. While the MMX instruction set does not provide a rounding mode, it is easy to accomplish in software. Listing 2 provides an example.

Listing 2. Rounding Using MMX Technology

```
// RoundVal is ½ LSB of 16-bit result
RoundVal = 0x0000400000004000;
pmaddwd mm6,mm5                 ;2.30 number
paddd   mm6,RoundVal            ;round
psrad   mm6,15                  ;2.30 to 1.15
```

Since the values are represented in two's complement, this technique works with both positive and negative numbers. In our Dolby Digital decoder, rounding was used extensively.

Gain Ranging

Although it can contribute to noise modulation, a Dolby Digital decoder can use Gain Ranging [7] to allow block scaling for low-level signals to increase dynamic range. This was used in our implementation.

Additional Performance Enhancements

Frequency Domain Downmixing

Since the TDAC transform is a linear process, downmixing can be accomplished in the frequency domain. This reduces the number of transforms from the number of input channels from the Dolby Digital stream (2 to 5) to the number of output channels (2). However, the transform block sizes in Dolby Digital can change from 512 to 256 in the presence of transients [7]. It is not possible to downmix in the frequency domain for differing block sizes, so in this case an additional downmix stage is required after the TDAC transform to perform the remainder of the downmix in the time domain. The transform coefficients are 32-bits. Using the 32-bit adds in the MMX instruction set provides an efficient downmix.

Early Scaling

There are several factors in the scale factor of a particular channel: Dynamic Range Control, Gain Ranging, and Downmix Scaling. We found it computationally beneficial to perform this operation during denormalization, essentially combining scaling and denormalization into one operation. This is performed by adjusting all of the exponents and mantissas by a particular amount. We stored the exponents as 8-bit quantities (the range is only 5 bits in Dolby Digital) and used MMX technology 8-bit add instructions (PADDB) to scale 8 exponents at a time. The unpack instruction (PUNPCKLBW) was used to efficiently replicate the 8-bit scale value eight times across the 64-bit register.

When the values are scaled up front, then downmixing becomes a simple addition as opposed to a multiplication by a constant. Since the transform coefficients are represented in 32 bits, downmixing in the frequency domain is performed by 32-bit adds using the packed add (PADDD) instruction. This avoids 32-bit multiplies.

Exponent and Bit Allocation Reuse

A Dolby Digital stream only has exponents in an audio block when the encoder determines that they have changed enough to be resent. This is called exponent reuse. Therefore, if exponent reuse is in effect, it is more processor-efficient to save the exponents in an array and use the values from the array (as opposed to re-extracting the bits from the bit stream).

The bit allocation information is derived from the exponents. Therefore if exponent reuse is in effect, bit allocation may also be reused. Since recalculating the bit allocation information is computationally expensive, the bit allocation information should be saved in an array and reused if possible.

Results

Compared to an optimized version that does not use MMX technology, the processor speedup is about 1.5X for a two-channel, surround-compatible (also known as LtRt) downmix from 5.1 channel source material. For 5.1 channels of output, the speedup increases to about 1.8X.

Decoding a Dolby Digital stream consumes less than 8% of a Pentium® II processor running at 233 MHz. Figure 2 shows the processor requirements for several DVD audio tracks (5.1 channels, 384K bits/second, 48K samples/second, downmixed to LtRt, except for Twister which is two channels, 192K bits/second).

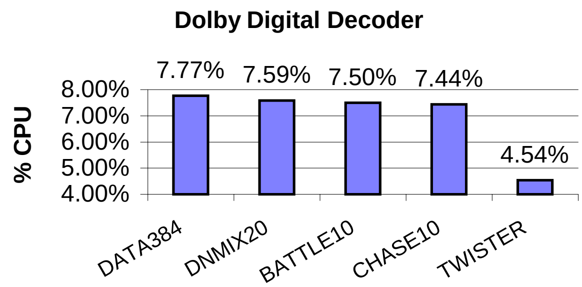

Figure 2. Processor Usage. BATTLE10 and CHASE10 are from the movie Outbreak. TWISTER is from the movie Twister.

The quality of Intel's Dolby Digital decoder compares favorably with floating-point based implementations. Intel's decoder has about 5 to 10 dB of additional noise as compared to a floating-point implementation. The improvement over a simple 16-bit truncation model is approximately 5 to 15 dB, depending on the program material. The most striking improvement is the reduction in peak spectral error, or the "tonality."

Figures 3 through 6 show how Intel's decoder compares to the 16-bit truncation model and floating-point reference.

Figures 3 illustrates the peak spectral error (graphical peaks) in the 300 to 20 kHz region. These peaks show the presence of tonal noise. Figure 4 shows the Total Harmonic Distortion (THD) vs. Frequency. The THD vs. Frequency is improved by about 10 dB over the 16-bit truncation decoder. Figure 5 is the noise modulation plot. This is a plot of the output noise in a third octave band at 4 kHz as a function of the input level of a 41 Hz sinusoid decremented from 0 dBFS to -120 dBFS. The improved (lowered) noise level is between 15 dB for high-level signals and 5 dB for low-level signals. Figure 6 is a noise plot of a 4 kHz sine wave reduced in level 1 dB per second from 0 dBFS to -120 dBFS, with the sine wave removed via a notch filter. This shows that the noise for a full-level signal is still small (-78 dB), going to -88 dB for a medium- to low-level signal. This is approximately a 12 dB improvement for high-level signals and approximately a 6 dB improvement for low-level signals.

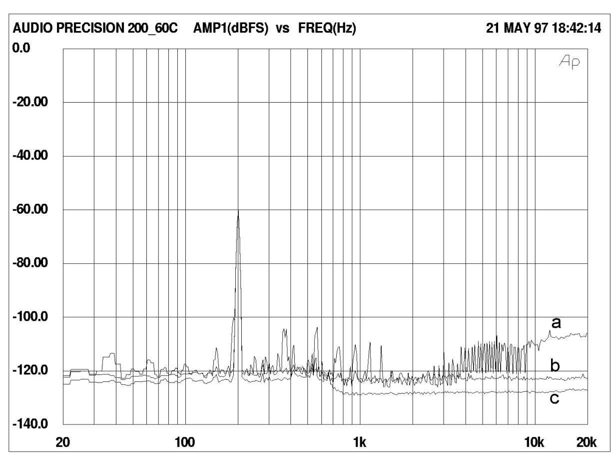

Figure 3. 200 Hz at -60 dB. a) 16-Bit Truncation, b) MMX Technology, c) Dolby Reference Decoder

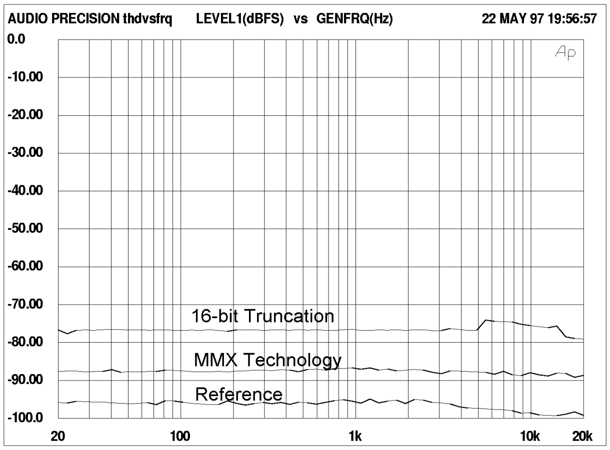

Figure 4. THD vs. Frequency

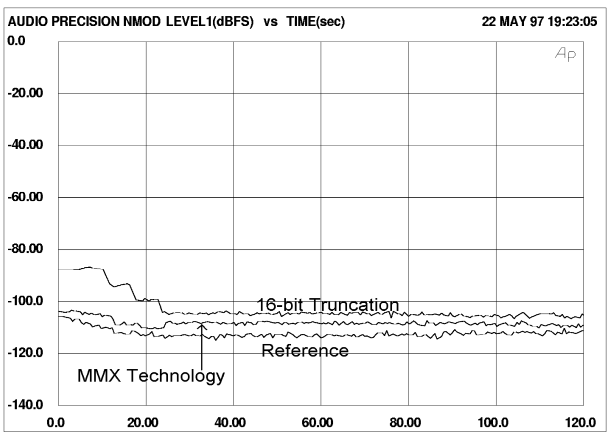

Figure 5. Noise Modulation at 4 kHz, 41 Hz Input

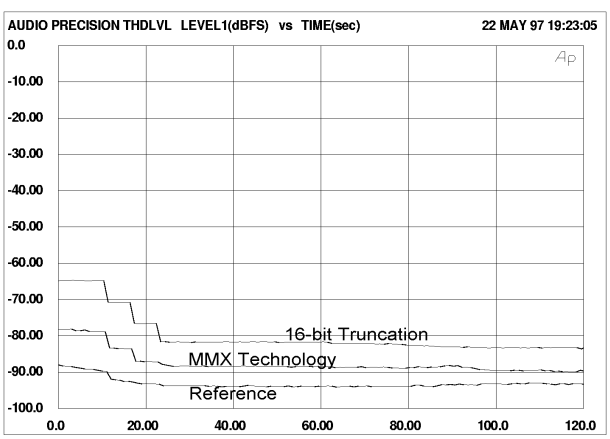

Figure 6. THD vs. Level, 4 kHz Input

Discussion

Table 1 shows the CPU breakdown for the decoder. After the data path has been sped up by MMX technology, the Bit Unpacking section becomes the next major consumer of the CPU. This is mainly due to the sequential nature of extracting variable-length bit fields from the bit stream.

Table 1. CPU Breakdown

Processing Block	% of Full Decoder
Bit Unpacking	28.3
TDAC/WOLA/Downmix	27.7
Scaling/Denormalization	27.2
Bit Allocation	10.2
Miscellaneous	6.6

Based on measurements (see Figure 6), the Intel decoder has a Signal-to-Noise Ratio (SNR) for a full-scale signal of about 78 dB. This compares reasonably well to the instantaneous sensitivity of the ear of about 85 dB [9]. The Dynamic Range (maximum output level vs. noise floor for a low-level signal) is about 88 dB. This compares reasonably well to a consumer CD player, which is typically at about 95 dB.

Conclusion

Intel’s Dolby Digital decoder provides a processor-efficient implementation that meets a high-quality standard. By offering this decoder as a baseline capability on PCs with MMX technology, decoding and playback of compressed audio is possible with no additional hardware cost. The low processor usage allows additional features such as software video decoding and audio enhancement to occur concurrently.

Acknowledgment

The writers thank Eric Benjamin, Charles Seagrave, Uttam Suri, and Steve Vernon of Dolby Laboratories for their suggestions on quality improvement and the help they provided in the analysis of the Intel decoder.

References

[1] “Using MMX™ Instructions to Get Bits From a Data Stream”, http://developer.intel.com/drg/mmx/appnotes.

[2] Sevic, D., Popovic, M., “A New Efficient Implementation of the Oddly Stacked Princen-Bradley Filter Bank,” IEEE Signal Processing Letters, Vol. 1, No. 11, pp. 166-168, Nov 1994.

[3] Davidson, G., Anderson, W., Lovrich, A., “A Low-Cost Adaptive Transform Decoder Implementation for High-Quality Audio,” IEEE International Conference on Acoustics, Speech and Signal Processing, Mar 23-26, 1992.

[4] Dolby Laboratories, “Test Procedure for AC-3 Decoders Using Test Files,” Dolby Document S96/11283.

[5] “Using MMX™ Instructions to Perform 16-Bit x 31-Bit Multiplication,” http://developer.intel.com/drg/mmx/appnotes.

[6] Princen, J. and Bradley, A., “Analysis/Synthesis Filter Back Design Based on Time Domain Aliasing Cancellation,” IEEE Transactions on ASSP, Vol. 34, pp. 1153-1161, Oct 1986.

[7] Advanced Television Systems Committee, “Digital Audio Compression Standard (AC-3),” Revision A/52, 20 Dec 1995.

[8] Thong, T., “Fixed-Point Fast Fourier Transform Error Analysis,” IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-24, No. 6, Dec 1976.

[9] Harris, S., “Understanding, enhancing, and measuring PC-audio quality,” EDN, Vol. 42, No. 8, Apr 10, 1997, p. 173.

Also see: http://developer.intel.com/technology/itj/q31997.htm

---