# Faster MPEG-1 Layer III Audio Decoding

*Scott B. Marovich*

Hewlett-Packard Company
Palo Alto, California 94304

*Abstract*

In MPEG-1 audio decoding, a method previously used to accelerate the synthesis subband filter's matrixing operation, plus a new extension of Lee's algorithm and other improvments, also accelerates the Layer III IMDCT step.

# Contents



# List of Figures



# List of Tables

# 1 Introduction

One of the most time-consuming steps in a naïve software MPEG-1 audio data decoder [1] is the synthesis subband filter's matrixing operation, which can be accelerated using Konstantinides' method [3]. Much of a Layer III decoder's overhead after making this improvement is spent in the preceding steps: an inverse modified discrete cosine transform (IMDCT), windowing, and overlapped addition of successive input vectors. Using Konstantinides' method and other improvements, we show that these can be similarly accelerated.

Recently Sakamoto *et al.* [7] noticed the trigonometric symmetry underlying our Theorem 1 but overlooked its connection to Konstantinides' method and consequent opportunities to further simplify IMDCTs. Liu and Lee's simplification by permuting vector elements [5] presents a result like Theorem 1 that also overlooks Konstantinides. Our derivation emphasizes this result's relation to previous work, and we pursue its consequences.

Section 2 follows Konstantinides' procedure to show how the IMDCT can be reduced to a fast DCT-like computation and some data copying operations. The cases of interest are 6- and 18-point IDCTs, which can be reduced to 3- and 9-point IDCTs using Lee's method [4], so Section 3 generalizes the latter to an odd number of points. Section 4 shows how to improve a decoder's windowing and overlapped addiition steps, then Section 5 analyzes our method's performance.

# 2 Exploiting Trigonometic Symmetry

Konstantinides [3] used a phase-shifting permutation of input vector elements to exploit trigonometric symmetry in the synthesis subband filter's matrixing operation, showing that this 32-to-64 element matrix multiplication can be reduced to a 32-point DCT-like computation, efficiently done using Lee's method [4] and 32 data copying or negation operations. We follow a similar procedure to decompose the IMDCT defined in [1] (*cf*. [5,7] for related results):

$$x_i = \sum_{k=0}^{N/2-1} X_k \cos\,(2i+1+N/2)(2k+1)\frac{\pi}{2N} \tag{1}$$

for $i = 0, \ldots, N-1$ and $N = 12$ or 36.

*Definition 1*. Let $x'_k$ be the phase-shifting permutation of $x_k$,

$$x'_i = \begin{cases} x_{i+3N/4}, & i = 0, \ldots, N/4-1 \\ x_{i-N/4}, & i = N/4, \ldots, N-1 \end{cases} \tag{2a}$$

whose inverse is:

$$x_i = \begin{cases} x'_{i+N/4}, & i = 0, \ldots, 3N/4-1 \\ x'_{i-3N/4}, & i = 3N/4, \ldots, N-1 \end{cases} \tag{2b}$$

Substituting (1) in (2a) yields:

$$x'_i = -\sum_{k=0}^{N/2-1} X_k \cos (2i+1)(2k+1)\frac{\pi}{2N},\ i = 0, \ldots, N/4-1 \tag{3a}$$

$$x'_i = \sum_{k=0}^{N/2-1} X_k \cos (2i+1)(2k+1)\frac{\pi}{2N},\ i = N/4, \ldots, N-1 \tag{3b}$$

*Lemma 1*. For $x'_i$ defined in (2),

$$x'_{N/2+i} = \begin{cases} -x'_{N/2-1-i},\ i = 0, \ldots, N/4-1 \\ x'_{N/2-1-i},\ i = N/4, \ldots, N/2-1 \end{cases} \tag{4}$$

*Proof*. For $i = 0, \ldots, N/4-1$, (3b) yields:

$$x'_{N/2+i} = \sum_{k=0}^{N/2-1} X_k \cos (N+2i+1)(2k+1)\frac{\pi}{2N} = -\sum_{k=0}^{N/2-1} -1^k X_k \sin (2i+1)(2k+1)\frac{\pi}{2N} \tag{5a}$$

$$x'_{N/2-1-i} = \sum_{k=0}^{N/2-1} X_k \cos (N-[2i+1])(2k+1)\frac{\pi}{2N} = \sum_{k=0}^{N/2-1} -1^k X_k \sin (2i+1)(2k+1)\frac{\pi}{2N} \tag{5b}$$

Comparing (5a) to (5b) proves the first case of (4). For $i = N/4, \ldots, N/2-1$, (3a) yields:

$$x'_{N/2-1-i} = -\sum_{k=0}^{N/2-1} X_k \cos (N-[2i+1])(2k+1)\frac{\pi}{2N} = -\sum_{k=0}^{N/2-1} -1^k X_k \sin (2i+1)(2k+1)\frac{\pi}{2N} \tag{5c}$$

Comparing (5a) to (5c) proves the second case of (4). □

Lemma 1 shows that we need only evaluate (3) for $i = 0, \ldots, N/2-1$, then do N/2 data copying operations.

*Definition 2*. Let $x''_i$ invert the signs of $x'_i$ as:

$$x''_i = \begin{cases} -x'_i,\ i = 0, \ldots, N/4-1 \\ x'_i,\ i = N/4, \ldots, N/2-1 \end{cases} \tag{6}$$

Note that (6) is its own inverse.

*Lemma 2*. For $x''_i$ defined in (6),

$$x''_i = \sum_{k=0}^{N/2-1} X_k \cos (2i+1)(2k+1)\frac{\pi}{2N},\ i = 0, \ldots, N/2-1 \tag{7}$$

*Proof*. For $i = 0, \ldots, N/4-1$, (6) and (3a) yield:

$$x''_i = -x'_i = \sum_{k=0}^{N/2-1} X_k \cos (2i+1)(2k+1)\frac{\pi}{2N} \tag{8a}$$

For $i = N/4, \ldots, N/2-1$, (6) and (3b) yield:

$$x_i'' = x_i' = \sum_{k=0}^{N/2-1} X_k \cos (2i + 1)(2k + 1)\frac{\pi}{2N} \qquad (8b)$$

□

Since (7) is an N/2-point DCT-like computation [2,6], Lemmas 1 and 2 establish:

*Theorem 1*. Output vector $\vec{x}$ of IMDCT (1) can be computed (within a scale factor) from the N/2-point DCT of $\vec{X}$ and N/2 data copying operations.

Like Lee [4], we convert (7) to a more easily computed form using the trigonometric identity,

$$2 \cos (2i + 1)\frac{\pi}{2N} \cos (2i + 1)(2k + 1)\frac{\pi}{2N} = \cos (2i + 1)k\frac{\pi}{N} + \cos (2i + 1)(k + 1)\frac{\pi}{N} \qquad (9)$$

Substituting (9) in (7) for $i = 0, \ldots, N/2-1$ yields:

$$2x_i'' \cos (2i + 1)\frac{\pi}{2N} = \sum_{k=0}^{N/2-1} X_k \cos (2i + 1)k\frac{\pi}{N} + \sum_{k=0}^{N/2-1} X_k \cos (2i + 1)(k + 1)\frac{\pi}{N} \qquad (10)$$

Subsituting $k = k' - 1$ in the second sum of (10), defining $X_{-1} = 0$, and observing that

$$\cos (2i + 1)k\frac{\pi}{N} \Big|_{k=N/2} = 0 \qquad (11)$$

for $i = 0, \ldots, N/2-1$, we rewrite (10) as:

$$x_i'' = \frac{1}{2 \cos (2i + 1)\frac{\pi}{2N}} \sum_{k=0}^{N/2-1} (X_{k-1} + X_k) \cos (2i + 1)k\frac{\pi}{N}, \; i = 0, \ldots, N/2-1 \qquad (12)$$

Expressing (12) using Lee's notation shows how (7) can be evaluated using his procedure:

$$G_i = g_i = 0 \qquad (13a)$$

$$H_k = X_{k-1} + X_k \qquad (13b)$$

$$h_i = \sum_{k=0}^{N/2-1} H_k \cos (2i + 1)k\frac{\pi}{N} \qquad (13c)$$

$$h_i' = \frac{h_i}{2 \cos (2i + 1)\frac{\pi}{2N}} \qquad (13d)$$

$$x_i'' = [g_i + ]\, h_i' \qquad (13e)$$

# 3 Fast Odd-Order IDCTs

Section 2 shows that for $N = 12$, (1) can be reduced to a 6-point DCT-like computation and 6 data copying or negation operations. A single application of Lee's method reduces this to two 3-point DCT-like computations of the form,

$$x_i'' = \sum_{k=0}^{2} X_k \cos (2i+1)k\frac{\pi}{6} \tag{14}$$

which expand to:

$$\begin{aligned} x_0'' &= X_0 + X_1\sqrt{3}/2 + X_2/2 \\ x_1'' &= X_0 - X_2 \\ x_2'' &= X_0 - X_1\sqrt{3}/2 + X_2/2 \end{aligned} \tag{15}$$

These are efficiently evaluated using two extra variables, 2 multiplications, and 4 additions, as shown by pseudo code in Figure 1.

$$\begin{aligned} T_0 &:= X_2/2 + X_0 \\ T_1 &:= X_1\sqrt{3}/2 \\ x_0'' &:= T_0 + T_1 \\ x_1'' &:= X_0 - X_2 \\ x_2'' &:= T_0 - T_1 \end{aligned}$$

**Figure 1.** Fast 3-Point IDCT Kernel

Section 2 shows that for $N = 36$, (1) can be reduced to an 18-point DCT-like computation and 18 data copying or negation operations. A single application of Lee's method reduces this to two 9-point computations like (12)-(13), and we follow a procedure like Lee's to decompose this case. Suppose that (12) includes a DCT-like computation of the form,

$$x_i'' = \sum_{k=0}^{N-1} X_k \cos (2i+1)k\frac{\pi}{2N},\ i = 0, \ldots, N-1 \tag{16}$$

where $N > 1$ is odd. Collecting even- and odd-index terms, rewrite (16) as:

$$x_i'' = \sum_{k=0}^{\lfloor N/2 \rfloor} X_{2k} \cos (2i+1)2k\frac{\pi}{2N} + \sum_{k=0}^{\lfloor N/2 \rfloor -1} X_{2k+1} \cos (2i+1)(2k+1)\frac{\pi}{2N},\ i = 0, \ldots, N-1 \tag{17}$$

Substituting $i = N-1-i'$ in (16) also yields:

$$x_{N-1-i}'' = \sum_{k=0}^{N-1} X_k \cos (2N - [2i+1])k\frac{\pi}{2N} = \sum_{k=0}^{N-1} -1^k X_k \cos (2i+1)k\frac{\pi}{2N},\ i = 0, \ldots, N-1 \tag{18}$$

which can be rewritten like (16) and (17) as:

$$x_{N-1-i}'' = \sum_{k=0}^{\lfloor N/2 \rfloor} X_{2k} \cos (2i+1)2k\frac{\pi}{2N} - \sum_{k=0}^{\lfloor N/2 \rfloor -1} X_{2k+1} \cos (2i+1)(2k+1)\frac{\pi}{2N},\ i = 0, \ldots, N-1 \tag{19}$$

Observe that (17) and (19) can be rewritten using Lee's notation with minor changes:

$$x_i'' = g_i + h_i', \; i = 0, \ldots, \lfloor N/2 \rfloor - 1 \tag{20a}$$

$$x_{\lfloor N/2 \rfloor} = g_{\lfloor N/2 \rfloor} \tag{20b}$$

$$x_{N-1-i}'' = g_i - h_i', \; i = 0, \ldots, \lfloor N/2 \rfloor - 1 \tag{20c}$$

$$G_k = X_{2k} \tag{20d}$$

$$g_i = \sum_{k=0}^{\lfloor N/2 \rfloor} G_k \cos (2i+1)k\frac{\pi}{N}, \; i = 0, \ldots, \lfloor N/2 \rfloor \tag{20e}$$

$$h_i' = \sum_{k=0}^{\lfloor N/2 \rfloor - 1} X_{2k+1} \cos (2i+1)(2k+1)\frac{\pi}{2N}, \; i = 0, \ldots, \lfloor N/2 \rfloor - 1 \tag{20f}$$

For $i = \lfloor N/2 \rfloor$, (20e) reduces to:

$$g_{\lfloor N/2 \rfloor} = \sum_{k=0}^{\lfloor N/2 \rfloor} -1^k \, G_k \tag{21}$$

The reader may verify that (20f) reduces to $h_0' = X_1 \sqrt{3}/2$ for $N = 3$, and that (20a)-(20e) then yield (15). For odd $N > 3$, substituting (9) in (20f) yields:

$$2h_i' \cos (2i+1)\frac{\pi}{2N} = \sum_{k=0}^{\lfloor N/2 \rfloor - 1} X_{2k+1} \cos (2i+1)k\frac{\pi}{N} + \sum_{k=0}^{\lfloor N/2 \rfloor - 1} X_{2k+1} \cos (2i+1)(k+1)\frac{\pi}{N},$$
$$i = 0, \ldots, \lfloor N/2 \rfloor - 1 \tag{22}$$

Substituting $k = k' - 1$, defining $X_{-1} = 0$, and noting that $N = 2\lfloor N/2 \rfloor + 1$ in the second term of (22) further yield:

$$2h_i' \cos (2i+1)\frac{\pi}{2N} = \sum_{k=0}^{\lfloor N/2 \rfloor - 1} X_{2k+1} \cos (2i+1)k\frac{\pi}{N} + \sum_{k=0}^{\lfloor N/2 \rfloor} X_{2k-1} \cos (2i+1)k\frac{\pi}{N}$$
$$= -1^i \, X_{N-2} \sin (2i+1)\frac{\pi}{2N} + \sum_{k=0}^{\lfloor N/2 \rfloor - 1} (X_{2k-1} + X_{2k+1}) \cos (2i+1)k\frac{\pi}{N},$$
$$i = 0, \ldots, \lfloor N/2 \rfloor - 1 \tag{23}$$

Equation (23) can be rewritten using notation like Lee's for $i = 0, \ldots, \lfloor N/2 \rfloor - 1$:

$$H_k = X_{2k-1} + X_{2k+1} \tag{24a}$$

$$h_i = -1^i \, X_{N-2} \sin (2i+1)\frac{\pi}{2N} + \sum_{k=0}^{\lfloor N/2 \rfloor - 1} H_k \cos (2i+1)k\frac{\pi}{N} \tag{24b}$$

$$h_i' = \frac{h_i}{2\cos(2i+1)\frac{\pi}{2N}} \tag{24c}$$

Equations (20) and (24) establish:

*Theorem 2*. For odd $N > 3$, the N-point DCT-like computation (16) can be evaluated using the $\lceil N/2 \rceil$-point DCT-like computation (20e), the $\lfloor N/2 \rfloor$-point DCT-like computation in (24b), $N-1$ multiplications, and $\frac{5N-7}{2}$ additions.

When evaluating (1) for $N = 36$, Theorems 1-2 and Lee's method produce 5- and 4-point computations of the form:

$$g_i = \sum_{k=0}^{4} G_k \cos(2i+1)k\frac{\pi}{9},\ i = 0, \ldots, 4 \tag{25a}$$

$$h_i' = \sum_{k=0}^{3} H_k \cos(2i+1)k\frac{\pi}{9},\ i = 0, \ldots, 3 \tag{25b}$$

It is not obvious how to recursively decompose these further, but they can be quickly evaluated by factoring common subexpressions *ad hoc;* for example, (25a) expands to:

$$\begin{aligned}
g_0 &:= G_0 + G_1\cos\frac{\pi}{9} + G_2\cos\frac{2\pi}{9} + G_3/2 + G_4\cos\frac{4\pi}{9}\\
g_1 &:= G_0 + G_1/2 - G_2/2 - G_3 - G_4/2\\
g_2 &:= G_0 - G_1\cos\frac{4\pi}{9} - G_2\cos\frac{\pi}{9} + G_3/2 + G_4\cos\frac{2\pi}{9}\\
g_3 &:= G_0 - G_1\cos\frac{2\pi}{9} + G_2\cos\frac{4\pi}{9} + G_3/2 - G_4\cos\frac{\pi}{9}\\
g_4 &:= G_0 - G_1 + G_2 - G_3 + G_4
\end{aligned} \tag{26}$$

These can be efficiently evaluated using three extra variables, 11 multiplications, and 15 additions, as shown in Figure 2.

$$\begin{aligned}
T_0 &:= G_3/2 + G_0\\
T_1 &:= G_0 - G_3\\
T_2 &:= G_1 - G_2 - G_4\\
g_0 &:= T_0 + G_1\cos\frac{\pi}{9} + G_2\cos\frac{2\pi}{9} + G_4\cos\frac{4\pi}{9}\\
g_1 &:= T_2/2 + T_1\\
g_2 &:= T_0 - G_1\cos\frac{4\pi}{9} - G_2\cos\frac{\pi}{9} + G_4\cos\frac{2\pi}{9}\\
g_3 &:= T_0 - G_1\cos\frac{2\pi}{9} + G_2\cos\frac{4\pi}{9} - G_4\cos\frac{\pi}{9}\\
g_4 &:= T_1 - T_2
\end{aligned}$$

**Figure 2.** Fast 5-Point IDCT Kernel

Similarly (25b) expands to:

$$
\begin{aligned}
h_0 &:= H_0 + H_1 \cos\frac{\pi}{9} + H_2 \cos\frac{2\pi}{9} + H_3/2 \\
h_1 &:= H_0 + H_1/2 - H_2/2 - H_3 \\
h_2 &:= H_0 - H_1 \cos\frac{4\pi}{9} - H_2 \cos\frac{\pi}{9} + H_3/2 \\
h_3 &:= H_0 - H_1 \cos\frac{2\pi}{9} + H_2 \cos\frac{4\pi}{9} + H_3/2
\end{aligned}
\tag{27}
$$

These are efficiently evaluated using two extra variables, 8 multiplications, and 10 additions, as shown in Figure 3.

$$
\begin{aligned}
T_0 &:= H_3/2 + H_0 \\
T_1 &:= H_1 - H_2 \\
h_0 &:= T_0 + H_1 \cos\frac{\pi}{9} + H_2 \cos\frac{2\pi}{9} \\
h_1 &:= T_1/2 + H_0 - H_3 \\
h_2 &:= T_0 - H_1 \cos\frac{4\pi}{9} - H_2 \cos\frac{\pi}{9} \\
h_3 &:= T_0 - H_1 \cos\frac{2\pi}{9} + H_2 \cos\frac{4\pi}{9}
\end{aligned}
$$

**Figure 3.** Fast 4-Point IDCT Kernel

# 4 Windowing and Overlapped Addition

For the most frequent input vector formats in [1], Type 0 blocks where $N = 36$ and Type 2 blocks of triples where $N = 12$, $x_i$ from (1) are windowed through the quantized half-sinusoid,

$$
z_i = x_i \sin (2i + 1)\frac{\pi}{2N},\ i = 0,...,N-1 \tag{28}
$$

and the first N/2 $z_i$ of one IMDCT are added to the last N/2 $z_i$ of the previous IMDCT. The 90° phase-shifting permutation (2) inhibits trigonometric symmetry to coalesce (28) and (1), but implementation-dependent simplification might be possible. If a host computer has a **multiply and accumulate** instruction, computations $z_i = s_i + x_i\ sin...$ can be coalesced at no extra cost. If the computer has enough registers to hold N/2 $x_i''$ from (7), one can eliminate some data copying and negation in (2b) and the inverse of (6) by windowing and overlapping $x_i$ in successive halves. To see how, note that (2b), (4), and the inverse of (6) map $x_i''$ to $x_i$ by quartiles of N as:

$$x_i = \begin{cases} x''_{i+N/4} & , i = 0, \ldots, N/4-1 \\ -x''_{3N/4-1-i} & , i = N/4, \ldots, N/2-1 \\ -x''_{3N/4-1-i} & , i = N/2, \ldots, 3N/4-1 \\ -x''_{i-3N/4} & , i = 3N/4, \ldots, N-1 \end{cases} \tag{29}$$

Linearly translating these quartiles' indices to the common input range, $i = 0, \ldots, N/4-1$, rewrite (29) as:

$$x_i = x''_{i+N/4} \tag{30a}$$

$$x_{N/2-1-i} = -x''_{i+N/4} \tag{30b}$$

$$x_{3N/4-1-i} = -x''_i \tag{30c}$$

$$x_{3N/4+i} = -x''_i \tag{30d}$$

These equations represent the same symmetries as [3; Fig. 1]. Note that (30a) and (30b) yield just the $x_i$ needed for a current IMDCT's windowing and overlap, and they access only the last N/4 $x''_i$ from (7), possibly in the opposite order. Likewise (30c) and (30d) yield (after windowing) just the $x_i$ to save for the next IMDCT's overlap, and they access only the first N/4 $x''_i$ from (7), again possibly in the opposite order. Thus the IMDCT, windowing, and overlap can be interleaved in two phases without extra copying. Moreover, negations in (30c) and (30d) can be factored out and done once.

# 5 Computation Time

Evaluating (1) naïvely requires $N^2/2$ multiplications and $N^2/2-N$ additions: 72 and 60 operations for $N = 12$, or 648 and 612 operations for $N = 36$. To compare our approach, note that Lee's method and the decompositions introduced here can be considered recursive functions on matrices that are composed in one order when evaluating (1) for $N = 12$ but in a different order for $N = 36$. A simple way to compute the number of multiplications and additions in each case is: (i) associate two scalar recursive functions with each decomposition primitive, tallying the two operations for each recursion level; (ii) compose each operation's tally functions isomorphically to the matrix functions' decomposition; (iii) evaluate the two scalar functional compositions for $N = 12$ and 36.

*Definition 3*. For Lee's method, let the tally functions producing [4; Table I] be:

$$\times_{LEE}(N) = \begin{cases} 2 \times_{?}(N/2) + N/2 & , N > 1 \\ 0 & , N = 1 \end{cases} \tag{31a}$$

$$+_{LEE}(N) = \begin{cases} 2 +_{?}(N/2) + 3N/2 - 1 & , N > 1 \\ 0 & , N = 1 \end{cases} \tag{31b}$$

For sign-inversion functions (4) and (6), let the joint tally functions be:

$$\times_{SGN}(N) = \times_?(N) \tag{32a}$$

$$+_{SGN}(N) = +_?(N) + N/2 \tag{32b}$$

For IMDCT (12) of Theorem 1, let the tally functions be:

$$\times_{TH1}(N) = \times_?(N/2) + N/2 \tag{33a}$$

$$+_{TH1}(N) = +_?(N/2) + N/2 - 1 \tag{33b}$$

Noting that $N = 2\lfloor N/2 \rfloor +1$ for odd-order DCT-like computations (20) in Theorem 2, let the tally functions be:

$$\times_{TH2}(N) = \times_?(\frac{N+1}{2}) + \times_?(\frac{N-1}{2}) + N - 1 \tag{34a}$$

$$+_{TH2}(N) = +_?(\frac{N+1}{2}) + +_?(\frac{N-1}{2}) + \frac{5N-7}{2} \tag{34b}$$

For minimal DCT-like computations (14) and (25), let the tally functions be:

$$\times_{DCT}(N) = \begin{cases} 2, & N = 3 \\ 8, & N = 4 \\ 11, & N = 5 \end{cases} \tag{35a}$$

$$+_{DCT}(N) = \begin{cases} 4, & N = 3 \\ 10, & N = 4 \\ 15, & N = 5 \end{cases} \tag{35b}$$

For $f = \times$ or $+$, compose (31)-(35) as:

$$f_{SGN} \circ f_{TH1} \circ f_{LEE} \circ f_{DCT}(3),\ N = 12 \tag{36a}$$

$$f_{SGN} \circ f_{TH1} \circ f_{LEE} \circ f_{TH2} \circ [f_{DCT}(5) + f_{DCT}(4)],\ N = 36 \tag{36b}$$

Here (32b) assumes that negation in (6) counts as "addition" but that data copying is free. Table 1 summarizes arithmetic operations in our method, from (36).

**Table 1.** Number of Arithmetic Operations

| N | × | | + | |
|---|---|---|---|---|
| | Old | New | Old | New |
| 12 | 72 | 13 | 60 | 27 |
| 36 | 648 | 81 | 612 | 149 |

For $N = 36$, the most frequent case in practice, our method eliminates 88% of the multiplications and 76% of the additions in a naïve evaluation of (1). For $N = 12$, the savings are 82% and 55%.

## 6 Conclusions

In a software MPEG-1 audio data decoder, Konstantinides' method of accelerating the synthesis subband filter's matrixing operation can also accelerate the Layer III IMDCT step. Together with a Lee-style method of performing DCT-like computations on an odd number of points, we have shown how to reduce the IMDCT's execution time by up to 88% and enable further implementation-dependent optimizations. Konstantinides sought to reduce the largest source of overhead in a typical decoder, and we build on his work to reduce the largest remaining source. These results let faster MPEG-1 Layer III audio decoders run on general-purpose computers.

## 7 References

1. Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbit/s; Part 3, Audio. ISO/IEC Standard No. 11172-3, 1994.

2. Jain, Anil K. A Sinusoidal Family of Unitary Transforms. *IEEE Transactions on Pattern Analysis and Machine Intelligence 1,* 4 (October 1979), 356-365.

3. Konstantinides, Konstantinos. Fast Subband Filtering in MPEG Audio Coding. *IEEE Signal Processing Letters 1,* 2 (February 1994), 26-29.

4. Lee, Byeong Gi. A New Algorithm to Compute the Discrete Cosine Transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing 32,* 6 (December 1984), 1243-1245.

5. Liu, Chi-Min, and Wen-Chieh Lee. A Unified Fast Algorithm for Cosine Modulated Filter Banks in Current Audio Coding Standards. *Journal of the Audio Engineering Society 47,* 12 (December 1999), 1061-1075.

6. Rao, K. R. and P. Yip. *Discrete Cosine Transform; Algorithms, Advantages, Applications.* (New York: Academic Press) 1990.

7. Sakamoto, Tadashi, Maiko Taruki, and Tomohiro Hase. A Fast MPEG-Audio Layer III Algorithm for a 32-Bit MCU. *IEEE Transactions on Consumer Electronics 45,* 3 (August 1999), 986-993.