MP3デコーダ IMDCT処理に必要な精度の評価
Evaluation of Precision Necessary for MP3 Decoder IMDCT Processing

神山智章 †　　清水尚彦 ††
† 東海大学大学院工学研究科
†† 東海大学電子情報学部コミュニケーション工学科

抄録

MP3 デコーダの IMDCT 処理を設計するに当たり、この処理において必要とされる精度について評価を行った。この結果に基づき、高速かつコンパクトな IMDCT 回路を実現できた。

Tomoaki KOUYAMA†　　Naohiko SHIMIZU††
†Graduate School of Engr. Tokai Univ.
††Dept. Comm. Engr., School of Information Technology and Electronics. Tokai Univ.

Abstract

I evaluated necessary precision of IMDCT processing for designing IMDCT circuit of MP3 decoder. I was able to realize IMDCT circuit that is compact and speedy.

1　はじめに

近年、普及している音声圧縮技術のひとつとして MP3 があげられる。この規格は ISO/IEC[1] で MPEG1 オーディオのレイヤ 3 として定められている。この規格は、非常に高音質でありながら、もとの PCM データの 10 分の 1 以下にまで圧縮ができる等の多くの利点がある。

しかし、この高音質高圧縮を実現するために非常に多くの計算を必要とする。そのため、エンコード、デコードともに高速な処理系を用いなければ実現することは非常に困難となる。特に、デコーダではリアルタイムでの動作を必ず要求される。

これらの処理を、ソフトウエア上で実現するためには非常に高速なプロセッサが必要となる。しかし、高速なプロセッサは消費電力が非常に多く、高価である。MP3 が主に用いられる組み込みの分野においては、そのようなプロセッサを用いることは困難である。

この問題の解決法として、専用ハードウェアを利用する方法が考えられる。この場合少ない回路サイズ、消費電力で必要な処理速度を得ることができる。

この専用ハードウェアを設計するとき、その処理に必要十分な計算精度が分かれば、さらに回路サイズなどを削減することができる。

今回 MP3 デコーダのための IMDCT 回路の設計にあたり、この処理内で必要とされる計算精度について評価を行い、実際に IMDCT 回路の設計を行ったので報告する。

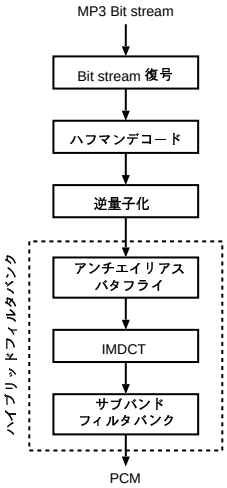

図 1: MP3 デコード処理

2　MP3 デコーダ

MP3 デコーダの構成を図 1 に示す。これらの処理は 1 グラニュール 576 サンプルを単位として行われる。

Bit stream 復号 MP3 Bit stream からデコードに必要な情報を抽出する。

評価値	デコーダの性能
8.810e-06 以下	Fully Compliant
1.410e-04 以下	Limited Accuracy
1.410e-04 以上	Not Compliant

表 1: Compliance Test

ハフマンデコード エンコード時に用いたハフマンテーブルで、デコードを行う。

逆量子化 MP3 Bit stream から取り出したパラメータを基に、ハフマンデコードされたデータを伸長する。

ハイブリッドフィルタバンク この処理は以下の 3 つの処理から構成される。

アンチエイリアスバタフライ エンコード時に削減された折り返しひずみを復元する。

IMDCT

サブバンドフィルタバンク 32 のサブバンドに分解された音声信号を、合成し復元する。

3 Compliance Test

MP3 デコーダの規格には、デコーダの音質に関して評価方法が Compliance Test として、定められている。

この評価方法は、リファレンスのデコードデータと、設計したデコーダの出力を比較するものである。比較に用いる計算式を以下に示す。

$$\sqrt{\frac{1}{N}\left(\sum_{i=1} N\left(t_i - r_i\right)^2\right)} \tag{1}$$

ここで、N:サンプル数、t_i:リファレンスデータ、r_i:デコードデータである。ただし、t_i、r_i は -1.0 から $+1.0$ の間で振幅を取る。

この式により得られた値がデコーダの性能を示す。この値は表 1 に示す、3 段階に分けられる。デコーダの要求仕様は、Limited Accuracy 以上の音質である。

4 IMDCT 処理

今回はこの IMDCT 処理部分について評価、設計を行った。

MP3 のデコードはグラニュール単位で行われる。1 グラニュールは、18 サンプルから成る 32 個のサブバンドから構成されている。この構成を図 2 に示す。

IMDCT 処理は、各サブバンド単位で処理が行われる。

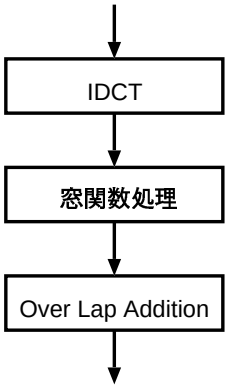

図 3: IMDCT 処理

IMDCT 処理は図 3 に示す様に、大きく、IDCT、窓関数処理、Over Lap Addition の 3 つの部分に分けることができる。それぞれの処理について詳しく説明する。

4.1 IDCT 処理

この IDCT 処理はブロックタイプによって異なる処理になる。

4.1.1 Long block

Long block の場合の IDCT 処理は次の式で与えられる。

$$x_i^{sb} = \sum_{k=0}^{\frac{n}{2}-1} X_k^{sb} \cos\left(\frac{\pi}{2n}\left(2i+1+\frac{n}{2}\right)(2k+1)\right) \tag{2}$$
$$\left(i = 0, ..., n-1 \quad n = 36 \quad X_k^{sb} : input \quad x_i^{sb} : output\right)$$

4.1.2 Short block

Short block の場合、サブバンドを更に 3 分割し、それぞれに対して IDCT 処理を行う。サブバンドはまず次のように分割される。

$$X_i^{sb,j} = X_{3i+j}^{sb} \qquad (i = 0, ..., 5 \quad j = 0, 1, 2)$$

IDCT 処理は分割されたものそれぞれについて次のように行われる。

$$x_i^{sb,j} = \sum_{k=0}^{\frac{n}{2}-1} X_k^{sb,j} \cos\left(\frac{\pi}{2n}\left(2i+1+\frac{n}{2}\right)(2k+1)\right)$$
$$(i = 0, ..., n-1 \quad j = 0, 1, 2 \quad n = 12$$
$$X_k^{sb} : input \quad x_i^{sb} : output) \tag{3}$$

a

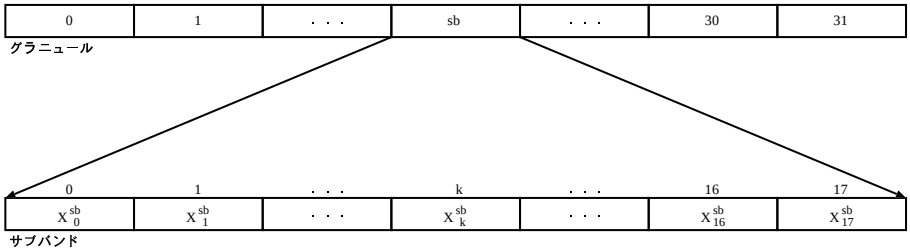

図 2: グラニュールとサブバンドの構成

4.2 窓関数処理

窓関数処理についても、ブロックタイプにより処理が異なる。

4.2.1 Long block

Long block は更にノーマルブロック、スタートブロック、ストップブロックの 3 つのタイプに分かれる。それぞれ別々の窓関数処理を行う。

- ノーマルブロック

$$z_i^{sb} = x_i^{sb} \sin\left(\frac{\pi}{36}\left(i+\frac{1}{2}\right)\right) \quad (i = 0, ..., 35) \tag{4}$$

- スタートブロック

$$z_i^{sb} = \begin{cases} x_i^{sb} \sin\left(\frac{\pi}{36}\left(i+\frac{1}{2}\right)\right) & (0, ..., 17) \\ x_i^{sb} & (18, ..., 23) \\ x_i^{sb} \sin\left(\frac{\pi}{12}\left(i-18+\frac{1}{2}\right)\right) & (24, ..., 29) \\ 0 & (30, ..., 35) \end{cases} \tag{5}$$

- ストップブロック

$$z_i^{sb} = \begin{cases} 0 & (0, ..., 5) \\ x_i^{sb} \sin\left(\frac{\pi}{12}\left(i-6+\frac{1}{2}\right)\right) & (6, ..., 11) \\ x_i^{sb} & (12, ..., 17) \\ x_i^{sb} \sin\left(\frac{\pi}{36}\left(i+\frac{1}{2}\right)\right) & (18, ..., 35) \end{cases} \tag{6}$$

4.2.2 Short block

ショートブロックの窓関数処理は 2 ステップに分かれている。まず、3 つに分解したものそれぞれについて以下の計算を行う。

$$y_i^{sb,j} = x_i^{sb,j} \sin\left(\frac{\pi}{12}\left(i+\frac{1}{2}\right)\right) \quad (i = 0, ..., 11 \quad j = 0, 1, 2) \tag{7}$$

次に、分解された 3 つを次のように合成する。

$$z_i^{sb} = \begin{cases} 0 & (0, ..., 5) \\ y_{i-6}^{sb,0} & (6, ..., 11) \\ y_{i-6}^{sb,0} + y_{i-12}^{sb,1} & (12, ..., 17) \\ y_{i-18}^{sb,2} + y_{i-12}^{sb,1} & (18, ..., 23) \\ y_{i-18}^{sb,2} & (24, ..., 29) \\ 0 & (30, ..., 35) \end{cases} \tag{8}$$

4.3 Over Lap Addition

この処理は前グラニュールの z_{18}^{sb} から z_{35}^{sb} と、現在のグラニュールの z_0^{sb} から z_{17}^{sb} の加算を行う。

直前のグラニュールのデータを $z_i'^{sb}$ とし、現在のグラニュールのデータを z_i^{sb} とすると次のような式になる。

$$result_i^{sb} = \begin{cases} -\left(z_{i+18}'^{sb} + z_i^{sb}\right) \\ \quad (i = 0, ..., 17 \quad i, sb \text{ 共に奇数}) \\ z_{i+18}'^{sb} + z_i^{sb} \\ \quad (i = 0, ..., 17 \quad i, sb \text{ いずれかが偶数}) \end{cases} \tag{9}$$

5 IMDCT 処理の改良

IMDCT 処理のハードウェア化のため、演算量の削減としていくつかのことを行っている。

5.1 三角関数のテーブル化

全て事前に計算しテーブルとして管理する。これにより、三角関数の複雑な計算は不要となる。

IDCT 処理での cos 関数は次のようにテーブル化を

行う。

$$COS[i][k] = \cos\left(\frac{\pi}{2n}\left(2i+1+\frac{n}{2}\right)(2k+1)\right) \quad (10)$$

$$(11)$$

この COS テーブルはロングブロック、ショートブロックそれぞれに対して用意する。

窓関数処理で用いられる sin 関数についても同様である。

5.2 演算量の削減

cos テーブル $COS[i][k]$ の乗算がロングブロックの場合次のように行われる。COS テーブルについては後述する。MID は IDCT 処理の出力を表す。

$$MID[i] = \sum_{k=0}^{17} N[k] \times COS[i][k] \quad (12)$$

$$i = \{0, 1, ..., 35\} \quad (13)$$

この cos テーブル $COS[i][k]$ にはロングブロックの場合、次のような周期性がある。

$$COS[i][k] = -COS[17-i][k] \quad (14)$$

$$COS[i+18][k] = COS[15-i][k] \quad (15)$$

$$i = \{0, 1, 2, ..., 7, 8\}$$

この周期性から $MID[i]$ には次のような関係がある。

$$MID[i] = -MID[17-i] \quad (16)$$

$$MID[i+18] = MID[15-i] \quad (17)$$

$$i = \{0, 1, 2, ..., 7, 8\}$$

この関係より、$i = 9$ から 26 の間での $MID[i]$ さえ求めれば、残りはコピー、および符号の反転とコピーで同じ処理ができる。これにより、乗算、加算ともに 50%ずつ削減できる。また、テーブルのサイズも半分で済む。

ショートブロックの場合も同様の関係が成立ち、同様の手法を用いることにより、乗算、加算をやはり 50%ずつ削減でき、テーブルサイズも半分にすることができる。

その関係を以下に示す。

$$MID[i] = -MID[5-i] \quad (18)$$

$$MID[i+6] = MID[11-i] \quad (19)$$

$$i = \{0, 1, 2\}$$

6 IMDCT 処理に必要な精度の評価

6.1 データフォーマット

IMDCT 回路に限らず、その回路サイズ、計算速度、精度などに大きな影響を持つのは演算に用いるデータフォーマットである。もし、必要とされる演算精度が定められているなら、その精度を満たす、最も省コストなデータフォーマットが望ましい。

IMDCT 処理の場合、浮動小数点形式が用いられることが多い。浮動小数点形式の場合、データの精度は主に仮数部の Bit 数より定まる。そこで、様々に仮数部の Bit 数を変化させたときの、それぞれのデータフォーマットがどの程度の音質を持っているのか評価を行った。その方法と、結果について説明する。

6.2 評価方法

ベースとなる MP3デコード環境として、PARTHENON 研究会の提供する MP3 デコーダ SW/HW 協調シミュレーション環境を用いた。この環境は論理シミュレータ SECONDS を C 言語から用いるライブラリ runseconds を用いた、ハードウェアとソフトウェアを協調動作させることのできるものである。

この環境内で、浮動小数点形式の仮数部のビット数を制限し評価を行う。浮動小数点形式の仮数部は以下の様にしてビット数の制限を行った。

6.2.1 Bit 数の制限

浮動小数点数 f を、frexp 関数を用いて仮数部と指数部とに分離する。このとき、仮数部は 0.5 以上 1.0 未満に正規化される。ここで、仮数部を f_f、指数部を f_e、N は制限する bit 数である。

$$f'_f = int(f_f \times 2^{N+1}) \quad (20)$$

$$f''_f = int(f_f \times 2^{N+2})\&(0\text{x}0001) \quad (21)$$

$$f'''_f = f'_f + f''_f \quad (22)$$

ただし、正負毎にそれぞれ処理を行う必要がある。この処理により精度を Nbit に制限することができる。この処理を全ての演算直後に施す。これにより、ほぼ正確に各演算精度で処理を行うことができる。

6.3 音質の評価

各 bit 数の精度毎に、Compliance Test を用いて、音質を評価する。その結果を、図 4 に示す。

この結果から、デコーダとしての性能を満たすには、IMDCT 処理において仮数部の精度として 9bit 以上が必要であることが分かる。また、15bit を越えるあたりからは評価値が飽和する。これは、他の処理における劣

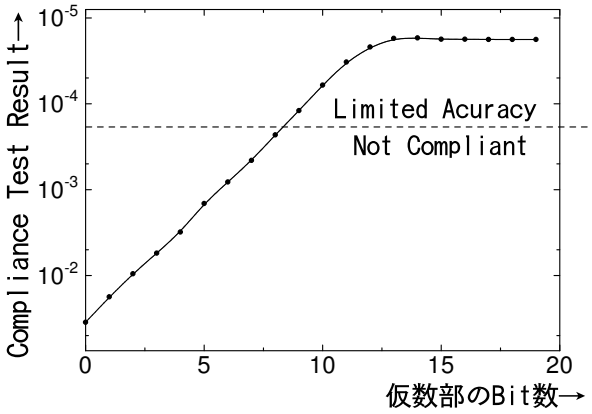

図 4: 仮数部の精度に対する音質の変化

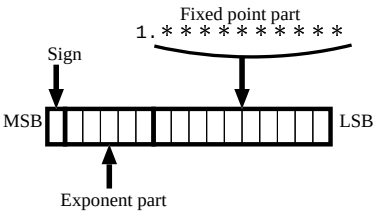

図 5: データフォーマット

化が IMDCT 処理の精度に対し十分でないためと考えられる。

7 検証

この評価方法で得られた、Compliance Test の評価値が実際に設計に活用できるものか、この結果に基づいて IMDCT 回路の設計を行った。

7.1 データフォーマット

処理に用いるデータフォーマットを浮動小数点形式とすると、図 4 の結果より、仮数部は 9bit あれば十分と言える。

しかし、演算器の回路サイズ縮小のためある程度精度は落ちると考えられる。そのため、今回は次のようなデータフォーマットを用いた。(図 5 を参照)

- 16bit 浮動小数点形式
- 符号:1bit (0:正、1:負)
- 指数部:5bit (0〜-63)
- 仮数部:10bit (暗黙の 1 あり、1.0 以上 2.0 未満)

また、指数部は、用いられるデータを集計すると 0 から −41 の間に分布することが判明した。よって、指数部を 5bit とし、指数の絶対値を取ったものを格納している。

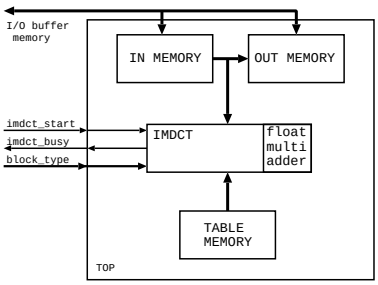

図 6: ハードウェアの構成

1.010266e-04	limited accuracy

表 2: compliance test の結果

7.2 浮動小数点演算器

この IMDCT 回路に用いる演算器については、回路サイズを優先する。そのため、内部で使用する整数演算器を 1 つのみとするなど、速度、精度的には厳しいものとなっている。

この演算器の主な構成要素を以下に簡単にまとめる。

- I/O レジスタ:17bit ×3 本
- 11bit 加算器 ×1
- 11bit シフタ ×1
- 4bit インクリメンタ ×1

7.3 IMDCT 回路

IMDCT 回路は SFL を用いて記述した。

IMDCT 回路のおおまかなハードウェア構成を図 6 に示す。

IMDCT 処理の内部は、基本的に、回路サイズ縮小のため複数の処理で同一のモジュールを共有するような構成となっている。

7.4 IMDCT 回路の評価結果

記述した回路について、MP3 デコーダ SW/HW 協調環境を用いて、シュミレーションを行い、その結果を Compliance Test によって評価した。

また、回路の論理合成にはセルライブラリとして NEC CMOS9 を用いて合成した。

compliance test の結果を表 2 に示す。この結果より、lmited accuracy を十分に満たしていることが分かる。

回路サイズ	遅延時間 [ns]	動作周波数 [MHz]
5450	18.58	53.82

表 3: imdct の合成結果

compliance test のグラニュール数	終了クロック
433[グラニュール]	48510518[クロック]
1 グラニュール当たりの平均	1 秒間に処理できるグラニュール
112034[クロック]	約 480 グラニュール/sec.

表 4: Imdct 回路の処理性能

また、合成結果を表 3 に、処理性能を表 4 に示す。回路は少ないゲート数で実現できており、かつ非常に高速に動作するものとなった。48kHz ステレオ音声をデコードする場合、1 秒当たり約 166 グラニュールの処理ができれば良い。設計した IMDCT 回路は、リアルタイムデコードに十分な処理速度を持っていると言える。

よって、図 4 に示した特性を参考にして、回路の設計は十分に可能である。

8 まとめ

本稿では、MP3 デコーダの IMDCT 回路において必要度される精度についての評価、検討を行った。その結果に基づき、実際に IMDCT 回路を設計した。シュミレーションでの評価において設計した回路は必要な性能を満たしていた。

今回は IMDCT 処理のみについて精度を落した評価を行ったため、他の処理での劣化によりこの精度では不十分になる可能性もある。しかし、デコード処理においては IMDCT 処理が大部分を占めるため、この評価値から大幅に外れることは無いと考えられる。

今後、デコーダの他の部分についての評価、設計、さらにエンコーダの IMDCT 処理についても検討して行く。

9 謝辞

MP3 デコーダ HW/SW 協調シミュレーション環境を提供して頂いた、京都大学の泉知論先生に感謝致します。

参考文献

[1] ISO/IEC 11172-3(Infomation technology-Coding of moving pictures and associated audio for digital strage media at up to about 1.5Mbit/s - part3:Audio)1993.

[2] http://www.8hz.com/mp3/

[3] 李 芝剛, 神山 智章, 清水 尚彦, 「MP3 エンコーダの高速化」, IPSJ Symposium Series Vol. 2001, No.16, pp.137-144, 2001 年 11 月

[4] 神山 智章, 李 芝剛, 清水 尚彦,「音声の統計データに基づいた MP3 エンコーダ量子化回路の設計と評価」, 第 19 回パルテノン研究会資料集, pp.39-56, 2001 年 11 月

[5] 小杉篤史
音声圧縮技術の基礎
Interface Aug.2001-Feb.2002 CQ 出版

[6] PARTHENON
http://www.kecl.ntt.co.jp/parthenon/