

Vorbis 音声伸張アルゴリズムの LSI 化とその評価*

6K-5

山根之訓

武内良典

今井正治†

北嶋暁‡

大阪大学大学院基礎工学研究科§

大阪電気通信大学総合情報学部情報工学科¶

1 はじめに

現在、音声圧縮伸張 LSI には低ハードウェアコストによる設計、特に低消費電力化設計が要求されており、汎用プロセッサによる情報圧縮伸張ではこの要求を満たすことができない。一方、特定用途向けプロセッサを用いて処理を行うことで、高機能化、高性能化、低電力化、低コスト化、高信頼化の効果が得られることが知られている [1]。

本稿では、Vorbis 圧縮伸張処理の中から変形逆離散コサイン変換 (IMDCT: Inverse Modified Discrete Cosine Transform) の専用回路化とその評価を行った。設計した演算器は動作周波数を下げることで低消費電力化を実現する。評価の結果、伸張処理を全て汎用プロセッサ (Pentium II) で行う場合と、IMDCT 処理を専用演算器で処理する場合を比較し、動作周波数を約 15% 低減できることが分かった。これより、Vorbis の専用回路化による低消費電力化の見込みを得た。

2 Vorbis

Vorbis は、MP3, AAC, TwinVQ 等と同様な音声信号の圧縮方式である [2]。Vorbis はその仕様が完全に公開されており、機能拡張提案も可能であるなど研究対象として適している。また、特許料フリーの特徴もある。

図 1 に Vorbis のブロック図を示す。Vorbis で用いられている圧縮方法は MP3 等と同じく人間の聴覚特性に注目した圧縮方法である。時間領域と周波数領域の変換には変形離散コサイン変換 (MDCT: Modified Discrete Cosine Transform) を用いている。音響心理モデルからの情報を元に MDCT のポイント数、量子化で割り当てられるビット数を動的に変化させる。

圧縮処理は、1. 入力信号の FFT による周波数解析 2. 音響心理モデルを用いた MDCT ポイント数、量子化ビットの決定 3. MDCT 4. 量子化 5. 符号化 6. ストリーム合成の順に行われる。

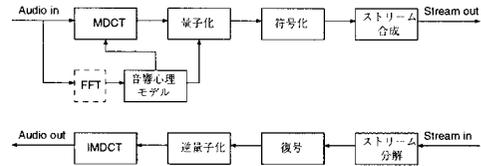


図 1 Vorbis のブロック図

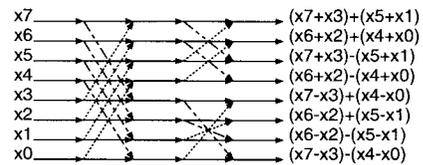


図 2 8 ポイントバタフライ演算

伸張処理は逆に、ストリーム分解、復号、逆量子化、IMDCT の順に行われ、音声を復元する。

3 LSI 化の検討

まず、伸張処理の LSI 化の検討を行う事とした。Vorbis 伸張処理のプロファイル結果から最も時間の必要となっている処理を最初に専用回路化することにした。プロファイル結果より伸張処理の中で最も時間の必要となる処理が IMDCT 処理であることが分かった。IMDCT は Vorbis 伸張処理のうち約 25% を占めていた。そこで、IMDCT 部の専用演算化を行うこととした。

次に処理の高速化を図るために、データ型の固定小数点化の検討を行った。浮動小数点で伸張した音と原音、固定小数点で伸張した音と原音との S/N 比を計測することで、符号に 1 ビット、整数に 3 ビット、小数に 12 ビットの計 16 ビットの固定小数点に決定した。

4 IMDCT 演算器の設計

IMDCT 演算では図 2 に示す 8 ポイントバタフライ演算が多用される。8 ポイントバタフライ演算は 16 回の加減算を行う演算である。図 2 において、左の $x_0 \sim x_7$ が入力で、右側が演算結果である。破線は加算、二重破線は減算を表している。本設計では、この 8 ポイントバタフライ用の専用演算器を設計した。

また、面積の増大を防ぐために加算、減算、乗算、積和演算を実行できる単一の演算器を作成した。この演算器は加算器と乗算器 1 つずつから構成される。

* VLSI implementation of IMDCT in Vorbis decoding and its evaluation

† Yukinori Yamane, Yoshinori Takeuchi, and Masaharu Imai

‡ Akira Kitajima

§ Department of Informatics and Mathematical Science, Graduate School of Engineering Science, Osaka University

¶ Faculty of Information Science and Arts, Department of Engineering Informatics, Osaka Electro-Communication University

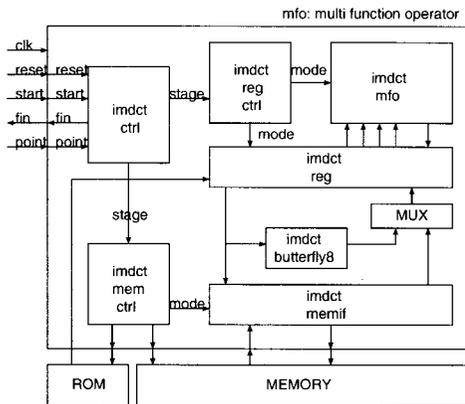


図3 IMDCT 演算器のブロック図

図3に設計したIMDCT演算器のブロック図を示す。設計したIMDCT演算器は、全体の制御部(imdct ctrl)、レジスタ制御部(imdct reg ctrl)、メモリ制御部(imdct mem ctrl)、レジスタ(imdct reg)、演算器(imdct mfo)、バタフライ演算器(imdct butterfly8)、メモリインターフェース(imdct memif)から構成される。

imdct ctrlは「メモリの読み込み」、「演算」、「メモリへ書き込み」の3状態の遷移を制御する。

メモリの読み書きはimdct mem ctrlがメモリとメモリインターフェースの制御を行い、読み書きが終了するとimdct ctrlに制御が戻る。演算時も同様にimdct reg ctrlがレジスタと演算器の制御を行い、演算が終了するとimdct ctrlに制御が戻る仕様となっている。

5 評価と考察

論理合成を行い、IMDCT演算器の面積と最大遅延時間を求めたところ、それぞれ100973ゲート、36.14nsとなった。表1に各コンポーネントの論理合成結果を示す。面積の大部分はレジスタによるものであり、レジスタ構成およびレジスタ数に改良の余地がある。

今回設計したIMDCT演算器と汎用プロセッサとを比較し、それぞれの必要とされる動作周波数について検討した。表2にIMDCT処理を汎用プロセッサとIMDCT演算器で行った場合の比較結果を示す。

汎用プロセッサでリアルタイム再生をするためにはIMDCT演算1回当たりの計算時間が3.03ミリ秒以下であればよく、57.3 MHz以上の動作周波数が必要である。設計したIMDCT演算器は汎用プロセッサと比較して約39%の実行サイクル数で演算を実行する。したがって、動作周波数を22.4 MHzまで低減できるこ

表1 合成結果

コンポーネント	面積 (gates)	最大遅延時間 (ns)
imdct_ctrl	7842	7.56
imdct_mem_ctrl	8682	17.63
imdct_reg_ctrl	17097	6.54
imdct_reg	42869	2.73
imdct_mfo	9353	34.41
imdct_butterfly8	5059	8.42
imdct_memif	2288	2.40
imdct	100973	36.14

論理合成ツール: Synopsys Design Compiler

ライブラリ: vsc753d (CMOS - 0.5 μ m)

表2 リアルタイムIMDCT処理のための要求

実装	a	b
汎用プロセッサ	174000	57.3 MHz
IMDCT 演算器	68000	22.4 MHz

a: 実行クロック数

b: リアルタイム再生に必要な動作周波数

とになる。

6 おわりに

本稿では、Vorbis専用プロセッサの設計を目標の第一段階として、伸張処理の中で最も時間の必要となるIMDCT演算部のLSI化を行った。シミュレーション結果から、ソフトウェア伸張処理のIMDCT演算を、今回設計を行ったIMDCT演算器に置き換えたときに平均動作周波数を15%程度低減することができた。

今後の課題として、(1) IMDCT以外の処理のハードウェア化、(2) モジュール間並列実行による動作周波数低減 (3) 圧縮処理部のLSI化があげられる。

謝辞

本研究に際し御討論いただいた大阪大学VLSIシステム設計研究室、株式会社アクセルの諸氏に感謝します。

参考文献

- [1] 今井正治 (編著), "ASIC技術の基礎と応用", 電子情報通信学会, 1994
- [2] "The Ogg Vorbis CODEC project", <http://www.xiph.org/ogg/vorbis/>
- [3] Th.Sporer, Kh.Brandenburg, B.Edler, "The use of multirate filter banks for coding of high quality digital audio", 6th European Signal Processing Conference (EUSIPCO), Amsterdam, June 1992, Vol.1, pages 211-214