社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2006－ＳＬＤＭ－123（１）
2006／1／17

FPGAを利用した高速オーディオフィンガープリントシステムの構築

磯永 久史† 井口 寧††

† 北陸先端科学技術大学院大学 情報科学研究科 〒923-1292 石川県能美市旭台1-1
†† 北陸先端科学技術大学院大学 情報科学センター, 科学技術振興機構 さきがけ研究21「機能と構成」
〒923-1292 石川県能美市旭台1-1
E-mail: †{h-isona,inoguchi}@jaist.ac.jp

あらまし 音楽や画像のデジタルコンテンツの膨大化に伴い、電子透かしやフィンガープリントなど、いくつかのコンテンツ管理の識別方式が提案されている。しかしながら、近年のファイル交換ソフトウェアによる、不正なネットワーク配信は著作権侵害やCD売り上げ数減少といった深刻な問題となっている。その対策として、例えば、ネットワーク上に流通しているコンテンツを識別するなど、コンテンツ管理技術をDRM(デジタル著作権管理)技術として利用する手法が注目されている。オーディオ用電子指紋(FingerPrint, 以下FP)とは、オーディオ内容の知覚に関連する部分をコンパクトに表現したものであり、たとえそれらが圧縮などの信号加工処理のため著しく質が劣化していたとしても、オーディオファイルを識別するために一番近いオーディオFPを使用できる。我々はオーディオFPアルゴリズムを改善することで高速なFPシステムをFPGA上に実現した。処理時間の評価では、ソフトウェアによる処理と比較して、約5.1倍の性能が得られた。

キーワード オーディオ用電子指紋，DRM(デジタル著作権管理)，FPGA，デジタルコンテンツ

Implimentation of high-speed audiofingerprint system using FPGAs

Hisashi ISONAGA† and Yasushi INOGUCHI††

† Japan Advanced Institute of Science and Technology Asahidai1-1, Nomi, Ishikawa 923-1292 Japan
†† Japan Advanced Institute of Science and Technology, "Infomation and Systems", PRESTO Japan Science and Technology Agency Asahidai1-1, Nomi, Ishikawa 923-1292 Japan
E-mail: †{h-isona,inoguchi}@jaist.ac.jp

Abstract Contents management technologies like the digital watermark and the fingerprint are proposed along with increasing of digital contents of music and pictures. However,In recent year illegal network deliveries with the file exchange software causes a serious problem such as copyright infringements and declining CD sales. In order to solve this problem, (for example,a digtal contents downloaded on the network are identified.etc)the method of applying content management technology to DRM(Digital Rights Management)technology is paid to attention. An audio fingerprint is a compact representation of the perceptually relevant parts of audiocontent. A suitable audio fingerprint can be used to identify audio files, even if they are severely degraded due to compression or other types of signalprocessing operations. We achieved a high-speed FP system on FPGA by improving the audio fingerprint algorithm. In the evaluation of the processing time, about 5.1 times the performance were obtained compared with processing with software.

Key words AudioFingerint , Digital Rights Management , FPGA , Digital Contents

1. まえがき

近年、音楽、映像などのデジタルコンテンツの膨大化により、ネットワーク上に流通している識別データを捕捉し、コンテンツの識別、管理を行うために、いくつかの流通情報管理の識別方式が提案されてきた。流通情報管理の識別方式の一つであるFPは、データ信号そのものの特徴を利用した、強固性が非常に強い技術である。データの特徴量を識別子としているため、データ部分への事前の著作情報埋め込みの必要が無く、ノイズに敏感な音楽ファイルなどに適している。オーディオFPの処

理の流れを図1に示す。FPはシステムに入力されたオーディオ信号の周波数帯域などの特徴量を解析し、算出された識別子を、データベースに問い合わせることで楽曲の検出処理を行う。しかし、FPは計算負荷が高い離散フーリエ変換などデジタル信号処理を用いて特徴解析をしなければならないので、検出時間が長く、他の識別方式と比較して読み取りコストが高くなる問題がある。したがって、ソフトウェアによる処理ではFP生

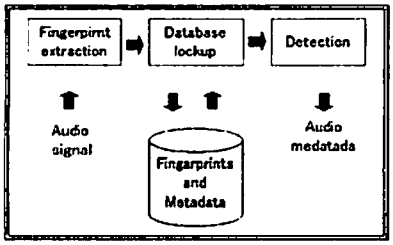

図1 フィンガープリントを利用した音楽の識別

成までの時間が長くなり、処理速度も遅くなる。したがって、ソフトウェアによる処理では、広帯域ネットワークで流通している膨大な数のデジタルコンテンツの補足・識別には対応することができないと考える。

そこで我々はデジタル信号処理を高速処理を得意とするハードウェアにより、オーディオFP生成回路の構築を行った。さらにパイプラインレジスタ処理を用いた並列処理や、モジュールレベルの並列展開による所要クロック数の削減などの高速化手法により、楽曲の識別にかかる処理時間の削減を目指した。

本稿の構成は、2章ではオーディオFPの概要とハードウェア実装向けアルゴリズムの概要を示し、3章ではハードウェア構築の概要、およびFP生成回路と楽曲識別回路の詳細を説明する。4章では構築したFPシステムの評価、ハードウェア処理とソフトウェア処理との比較および考察を行う。最後に5章でまとめと今後の予定を述べる。

2. オーディオフィンガプリントについて

2.1 従来研究

これまでのオーディオFPの従来研究に関しては、強度、利便性、算出オーダの短縮などに尽力が注がれてきた。Canoらは、プロトタイプとなるオーディオFPシステムの概要[1]を提案した。すべてのオーディオFPシステムは、それらのFP内に存在するいくつかの時間-周波数表現から導かれる。つまり、短フーリエ変換を利用する。FP算出の主な違いとは、それらがFPを構築するために使用する特徴によって決定される。つまり特徴とは、スペクトル扁平の特徴[2]、スペクトルのピークの特徴[3]、フーリエ係数の特徴[4]、メル周波数ケプストラム係数の特徴[5]、そして周波数帯間のエネルギー差の特徴[6]などである。

我々は高速かつ少回路量であり、信頼性の高いハードウェアFPシステムの構築を実現するために、MP3などの圧縮に対して、非常に強健であり、生成アルゴリズムのステップが加算と減算などの簡素な構成であること、回路量が膨らむ乗算および除算が比較的少ないこと、また生成されたFPサイズがコンパクトで、FPGA内で省スペース(FPGAの4input-LUT消費など)で格納・容易に検出ができること、などを考慮した結果、HaitsmaとKalkerらによる、Philipsオーディオ FPシステム[6]をハードウェア用に改善し、実装を行うことにした。

2.2 HaitsmaとKalkerrらの特徴生成アルゴリズム

図2にHaitsumaらが提唱したFPのシステムの生成ステージの概観を示す。最初に音声信号は、31/32のオーバラップされた要素である、0.37秒のフレームに分割され、Hanningウィンドにより重み付けられる。分割された、オーディオフレームのコンパクトな表現は、サブFPと呼ばれる。それは11.6ms(370ms)毎の間隔で、32ビットの「サブ-FP」を生成する。

大きいオーバラップのため、サブFPの系列は大きな類似性を持ち、時間内で少しづつ異なっていく。楽曲はサブFPの256個の連続した系列から構成され、データベースに格納される。それぞれのフレームに32ビットのサブFPを生成するために、33個の非重である周波数帯が見積もられたパワースペクトル密度(Power Spectral Density)から選出される。パワースペクトル密度は、簡単なperiodogram概算を使用することで見積もられる。これらの帯域の範囲は人の聴覚の特性と同じ300Hzから2000Hzと対数関数により区切られる。

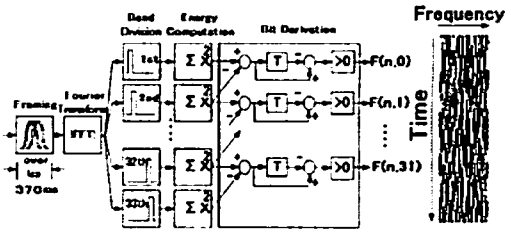

図2 左：Haitsumaらのオーディオ FP特徴生成 右:FP例 白→F(n,m)=1 黒→F(n,m)=0

HaitsmaとKalkerはエネルギー差のサインが多くの種類の処理に非常に強健な特性であることを示した[6]。

文献[6]の記法を使用して、ここではフレームnの周波数帯mのエネルギーをE(n、m)で記す。これらのエネルギー差ED(n,m)は時間と周波数で計算され、

$$ED_{(n,m)} = (E_{(n,m)} - E_{(n,m+1)}) - (E_{(n-1,m)} - E_{(n-1,m+1)}) \quad (1)$$

サブFPのビットF(n,m)は

$$F_{(n,m)} = \begin{cases} 1 & ED_{(n,m)} > 0 \\ 0 & ED_{(n,m)} \leq 0 \end{cases} \quad (2)$$

から算出され、式(2)のF(n、m)はnフレーム目の第mビットを意味する。図2の右側は実際のFPの例を示す。白い部分は、ポジティブなエネルギー差(すなわち、F(n、m)=1)を示して、黒い部分は、ネガティブなエネルギー差を示す。FPブ

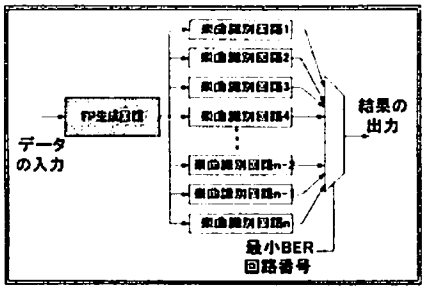

図 3　FP 生成回路の構成

表 1　実装環境

FPGA ボード	日立　Logic Bench
HW/SW インターフェイス	日立 Virtual Turbo PCI
搭載 FPGA	Xilinx XC2V6000FG1156-4×4
ホスト PC CPU	Athron 3.6GHz
ホスト PC OS	Windows2000 Pro
ホスト PC メモリ	1 GB
連動アプリケーション	Visual Studio.net
開発環境	Xilinx ISE6.3i
論理合成ツール	Synplicity Synplify Pro V7.3.4
開発言語	VHDL 、C++

ロックの短辺は、33 の周波数成分の間の違いに対応する、32 ビットから成る周波数帯の傾向であり、ブロックの長辺は時間軸に対応する。256 個のサブ FP(合計サイズ 1KB) を発生させて、たとえ楽曲のセグメントがさまざまな信号加工処理のために品質劣化しても、このシステムは、大容量データベースでわずか 3.3 秒の音楽のセグメントを識別することを可能とした。「生成された FP」と「データベースの FP」の間の Bit Error Rate(BER) は識別のための類似性測定のパラメータとして使用される。

BER が閾値の下にある場合は、照合値がデータベース項目で見つけられたということである。このシステムでは、閾値は 0.35 に設定される。

3.　ハードウェア構築について

3.1　ハードウェア構築の概要

本稿では、ハードウェアにおける、オーディオ FP システムの高速設計手法について説明する。我々が提案したハードウェアにより、高速化されるシステムは既存の FP システムと同様、FP 生成部と楽曲検索・識別部の 2 ステップ構成 (図 3 参照) とした。ここでの FP 生成部とは、図 2 の Haitsuma らの FP アルゴリズムを、ハードウェア向けに改良したものである。

ハードウェア部はホスト PC 上の C++プログラムと連動を行う。ホスト PC 上で FPGA ボードの専用の書き込み用 API 関数を実行することで、送信されてきた音楽ファイル (wave、16bit、44.1KHz) を入力とし、2 章で既に説明した Haitsuma らの生成アルゴリズムに準じて FP を生成する。FP 生成部は各演算部間へのパイプラインレジスタ挿入や比較器の並列展開などの手法により、性能の向上を目指した。楽曲識別回路には FPGA に標準搭載されている BRAM もしくは CLB によって構成される ROM を配置し、そのメモリ領域には、あらかじめ入力が予想される楽曲の FP を格納しておく。楽曲識別回路を並列に配置し、同時に複数の識別処理を行うことにより、楽曲識別の所要時間の削減を行う。

3.2　実装環境

オーディオ FP システムの実装環境を表 1 に示す。日立製の FPGA ボード「Logic bench」は、Xilinx の FPGA「XC2V6000」を 4 個搭載しており、ゲート数は 2400 万システムゲートと大規模論理検証が可能である。また「XC2V6000」は、BRAM を約 9Mbit 相当を含んでおり、この FPGA ボードのリソースを最大限に活用した場合は、少なくとも約 400 個の楽曲識別回路の構築が可能となり高並列での識別処理が可能である。ホスト PC と FPGA 間のブリッジは日立製「Virtual Turbo　PCI インターフェイスボード」を使用した。この PCI ボードは SW・HW 連動シミュレーションのための専用 API が用意されており、多様なシステム開発に適応できる。また、基幹クロックは 33MHz ではあるが、OSC の予備ソケットが搭載されているので、さらに高速なクロック周波数で動作させることも可能でであり、拡張性に優れている。

3.3　FP 生成部

FP 生成回路は 6 段パイプライン構成で図 2 の処理を行う。この 6 段のパイプラインの過程で使用する演算器は高速フーリエ変換器、加算器、減算器、符号なし乗算器、コンパレータなどである。パイプラインのビット幅は wave ファイル、PCM 方式の量子化 16bit に合わせる。このパイプラインは以下の 6 回のステージで考えることができる。

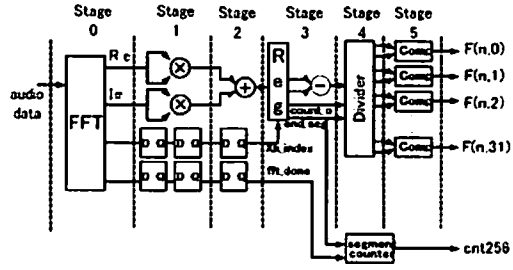

図 4　FP 生成回路の構成

[ステージ 0]:高速フーリエ変換

入力デジタル信号 (N=64 サンプル)$\{f_n\}_{n=0}^{n=63}$ から高速フーリエ変換 (式 (3)) を用いて実数部、虚数部各々の周波数成分 $\{F_k\}_{k=0}^{k=63}$ を算出する。

$$F_k = \frac{1}{N}\sum_{n=0}^{N-1} f_n \left(e^{-j(2\pi/N)}\right)^{nk} \quad (k = 0, 1 \cdots 63) \qquad (3)$$

ただし、$e^{-j(2\pi/N)} = cos(2\pi/N) - jsin(2\pi/N)$

この高速フーリエ変換部は、設計の効率化のため、Xilinx 社

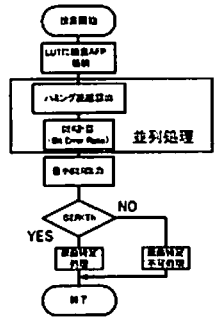

図 5　楽曲識別アルゴリズム

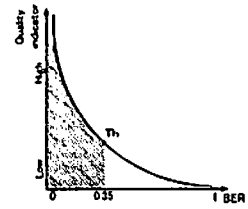

図 6　クオリティと BER の関係

が ISE 設計ツールで無償で提供する IP コアを活用した。これは Radix-Ⅱバタフライ演算と、いくつかのパイプラインレジスタで構成されているため、最適な回路量にて高速な演算を実現できるものである。

[ステージ 1]:乗算器

周波数成分の実数部と虚数部の 2 乗をそれぞれ計算する。

[ステージ 2]:加算器

実数部と虚数部、各々の 2 乗の和を計算する。

[ステージ 3]:パワースペクトル密度総和の計算

レジスタ内にて、1 セグメント分の各周波数成分の総エネルギー値 (PSD 総和) を算出する。高速フーリエ変換は 64 サンプル入力毎の周波数成分を出力するため、1 セグメント (13384 サンプル) 当たり、256 回繰り返し加算を行うことで各周波数成分の総エネルギーを算出し、その隣合う差を出力する。

[ステージ 4]:エネルギー差計算

ステージ 3 にて算出した 1 セグメント毎の各周波数成分の総エネルギー差を、32 個のコンパレータに分配する。コンパイラは前後のセグメントのエネルギー差を比較する仕様であるため、この divider には前セグメントのエネルギー差の値を記憶しておく回路が必要である。

[ステージ 5]:比較器並列展開

式 (2) の条件に従い、並列に配置したコンパイラにてサブ FP32bit を同時に算出する。

周波数成分の番号 index(図 4 参照) はステージ 3 の「Reg」で初めて必要となるため、3 段のパイプラインレジスタを経由することで、データと周波数成分の番号を同期させている。

また、パイプライン処理を効率的に行うためには、各ステージ間の機能粒度をほぼ均衡にする必要がある。ステージ 3 の「Reg」、ステージ 4 の「divider」はそれぞれモジュール内に複数のパイプラインレジスタを含む処理をすることでクリティカルパスの遅延を抑え、動作周波数の向上を実現する。

3.4　楽曲検索・識別部

3.4.1　Bit Error Rate(BER) について

FP システムにおいて、任意の 2 つの全く異なる楽曲では、全く異なる FP となるが、同一の曲でも片方が MP3 への圧縮などの加工により、質が劣化したものである場合、それらのオーディオ FP は原曲のものと完全一致するとは限らず、多少は異なる。そのような加工されたオーディオファイルでも識別できることがシステムとして望ましいと考えられ、ある程度の誤差も許容しなければならない。そこで、識別を決定するパラメー

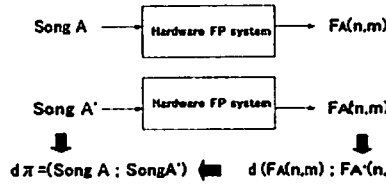

図 7　知覚的な音質の差と FP の差の関連性

タとして、BER(Bit Error Rate) を本研究では採用する。ここでは、原曲 A から生成された FP を $F_A(n,m)$、加工処理された曲 A' から生成されたものを $F'_A(n,m)$ とする。BER は比較する二つの FP $F_A(n,m)$ と $F'_A(n,m)$ のハミング距離

$$F_{diff}(n,m) = F_A(n,m) \oplus F_{A'}(n,m) \tag{4}$$

から算出される。したがって、ハミング距離に基づいた BER は

$$BER = \frac{1}{32N} \sum_{n=0}^{N-1} \sum_{m=0}^{31} F_{diff}(n,m) \quad (N = 1, 2 \cdots 256) \tag{5}$$

となる。この二つの FP のハミング距離は、曲 A と曲 A' の知覚・感性的な差 $d_\pi(SongA, SongA')$ と関係するとされている。つまり、BER の差は原曲と加工された曲との感性の違いを数値化したものであり、2 者間で BER が低い場合、類似性により同一楽曲であるという確率が非常に高くなる。

3.4.2　Bit Error Rate(BER) に基づいた楽曲識別法

本稿では、前項で述べた BER を利用した楽曲識別を FPGA 内で行う手法とその構成について説明する。我々が提案する楽曲識別アルゴリズムは図 5 に示されるとおりである。この楽曲識別手法の特徴として、図 3 に示すように楽曲識別回路を並列配置することで、「システムに入力された音楽データから生成された FP」と「データベースに格納されている FP」を比較対象とし、同時に複数のハミング距離、BER 演算を行う。そうすることで、高速な楽曲識別を可能とする。モジュールレベル（大規模な機能回路）の並列配置は、ファンアウトを拡大させクリティカルパスの遅延が大きくなる恐れがあるため、論理合成時

表 2　処理時間の比較
(クロック周波数=33MHz)

	FP 生成時間	楽曲識別時間	性能比
ソフトウェア	3353ms(19.95Mbps)	−	1.0
ハードウェア	650.24ms(102.78Mbps)	0.543ms	5.152

表 3　回路量とクリティカルパス

	4 input LUTs	critical path	max freq
FP 生成部	1997(67584)	11.500ns	86.957MHz
全体 (FP 生成部+楽曲識別部 10 並列+IF 部)	5107(67584)	25.727ns	39.956MHz
全体 (FP 生成部+楽曲識別部 30 並列+IF 部)	5888(67584)	26.138ns	38.258MHz

に出力されるレポートファイルの結果を参考にしながら、並列度数の調整を行う。楽曲識別は以下のステップで構成される。

- **生成 FP の格納**

FP 生成部にて生成された FP を LUT(Look Up Table) に格納する。LUT は RAM(データバス幅 32 ×アドレスバス幅 8) で構成し、N セグメント目の FP の格納アドレスは N-1 という規則性を持たせる。例えば 6 セグメント目の 32bit サブ FP はアドレス:0x05 に格納する。RAM は FPGA に標準搭載されているものを使用する。我々が実装を行う Xilinx 社の FPGA XC2V6000FG1156 は約 9Mbit の BRAM を含んでいるので、1 つの FPGA には最大 100 個 (8192bit=1KB/1FP) の FP データベースが蓄積ができる。

- **BER の並列計算**

並列処理にて同時に、複数の「データベース FP」と「問い合わせ FP」間の BER(式 (5)) を算出する。2 つの FP 間ハミング距離の算出は、アドレス:0x00 から開始し、0xFF で終了する。つまり 1 セグメント目から昇順にハミング距離を算出する。

- **最小 BER の出力**

複数の楽曲識別回路から出力された BER の比較を行い、最小の BER 値と、その FP データベース番号を出力する。ここではクリティカルパスの遅延が長くならないよう、2 分木構造の比較回路構成とした。

- **最小 BER と閾値の比較**

手動にて設定した閾値と、出力された最小の BER の比較を行う。Haitsuma らの文献 [6] では、BER が閾値=0.35 より下にある場合は、問い合わせ楽曲がデータベース中から発見できたということを実験にて証明している。本稿ではスレッショルド =0.35 で設定し、BER が閾値より下にある場合 (図 6 網掛け部) は楽曲識別フラグ、つまりデータベース番号を出力する。最小 BER が閾値より上にある場合は、特定不可フラグ ALL"1" を出力する。

4. 評　　価

4.1 ソフトウェアによるオーディオ FP

Haitsuma らの FP アルゴリズムをソフトウェアにて実装を行い、予備実験として FP を生成する時間を測定した。実装環境として、CPU は opteron プロセッサ 2GHz × 2、メモリ 2GB と高性能な計算機を利用した。また、OS は Linux Fedore core 3 を利用し、FP システムに入力するオーディオファイルは、wave ファイル、PCM 方式、16bit、44.1KHz とする。wave、左チャンネルの 256 セグメント分からオーディオ FP を生成するプログラムを C 言語で作成し、そのプログラム実行に要する時間を time コマンドにて測定を行った。

ユーザー CPU 時間 (user) の結果から、処理ビットレートは 19.95Mbps となった。周波数解析に高速フーリエ変換を用い、デュアルプロセッサで構成される高性能な計算機にて処理したにもかかわらず、高速な処理は不可能であった。またメモリ／CPU 間のデータ入出力など時間も含めると、実時間 (real) は約 7 秒となってしまう。したがって、ソフトウェアによる手法では局所的なコンテンツ識別には対応できるが、100Mbps クラスの広帯域なネットワーク上で音楽コンテンツの捕捉・識別するなどの応用は不可能である。

4.2 フィンガープリント処理時間の比較

次に、ハードウェアでの FP 処理時間の測定を行った。測定系を図 8 に示す。ホスト PC の C 言語にて、wave のヘッダファイルを取り除く処理を行い、wave の左チャンネルのデータ部を書き込み API を呼び出すことにより、PCI バス経由で FPGA ボードに転送する。FPGA 上の PCI-FPGA ブリッジは Vir-Tual Turbo の規定タイミングにリタイミングを行い、FPGA にデータを書き込む。この FP システムを今後、大規模なもの

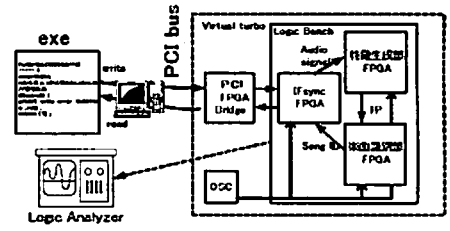

図 8　測　定　系

へ拡張するため、システムゲートの余裕を考え PCI-FPGA との同期を行う IF sync 部、特徴生成部、楽曲識別部をそれぞれ 3 つの FPGA に論理分割を行った。

事前準備として、それぞれの楽曲識別部には異なる 30 曲分の FP を蓄えておき、ホスト PC から音楽データ 1 曲を入力し、

楽曲識別を行う。時間測定は書き込みAPI実行から読み取りAPI終了までのclock関数を用いた時間測定、およびロジックアナライザのモニタにて、FP所要実時間測定を実施した。システムクロック33MHzで動作させた場合、wavファイル256セグメント(約8MB)のFPにかかる時間は約650ms(ビットレートに換算すると102Mbps)、楽曲識別は約0.54msという結果が得られた。ソフトウェアのFP生成速度と比較すると約5倍の性能を達成した。

ロジックアナライザでの解析によると、所要時間の約90%は高速フーリエ変換の演算時間である。入力信号64サンプルにつき、高速フーリエ変換の演算所要時間が約9.7 μ Sかかり、音楽データ256セグメントのFP生成には入力信号4177152サンプルの演算をしなければならないのでFFT合計時間は、少なくとも約630msかかる。

4.3 回路量とクリティカルパス

本稿で構築した10並列、30並列楽曲識別回路の消費回路量、およびクリティカルパスを表3に示す。30並列の回路においても4-input-LUTは5888と、少ない回路量で構築することができた。これは実装環境であるXC2V6000全体の回路量では10%しか消費していないことになる。したがって本稿で実装した環境においては楽曲識別回路の並列度数は回路量ではなくて、FPGAに含まれるRAM容量の上限にて制限される(最大100並列)。

また、10並列時のクリティカルパスは25.727nsとなり、FP生成部単体と比較して2倍以上になっている。これは楽曲識別部内にて、式(5)のBERを計算する際、VHDLのInteger変数の加算をloopすることでビットエラーの総数を求めるブロック、およびBER最小値を算出するブロックによるファンアウトの増大がクリティカルパスを著しく増加させている原因と考えられる。

しかし、10並列から30並列へ拡張した結果、クリティカルパスは0.4ns程度しか増加していないことから、楽曲識別回路の並列数はクリティカルパスの増加にはさほど影響しないことが分かった。つまり、33Mの動作周波数で動作させた場合は100程度の並列数では動作周波数は減少せず、並列数を増加させても処理速度が低下することはない。

5. まとめと今後の予定

本稿では以下の2点を考慮して高速なFP生成・識別回路を構築した。

- パイプライン処理によるFP生成回路の性能向上
- 楽曲識別回路の並列配置による楽曲同時検索

我々が構築したハードウェアは、オーディオデータから高速にFP生成を行い、複数の楽曲を同時検索することが可能である。30楽曲の同時識別を試みた場合では102Mbpsの速度で楽曲識別が可能となりソフトウェアによるFP処理と比較して約5.1倍の性能向上が得られた。これは広帯域なネットワーク上での音楽ファイルの捕捉が可能となるなど、コンテンツ流通・管理の処理能力向上という点で有益な結果が得られたと言って良い。

しかし、本稿で提案したシステムでは、少数のタイトルしか識別可能としないため、世の中に存在する膨大な音楽のタイトル数を考えた場合、楽曲のデータベース数に関して言えば容量不足である。HMVなどの大手レコード会社がインターネットで配信する音楽のタイトル数は、60万〜70万タイトルと増加する一途であり、本稿で提案したシステムをさらに実用的とするためには、FPGAと高速に通信可能な大規模な記憶媒体が必要である。

今後の予定として、インターネットで流れている音楽ファイルの捕捉・楽曲識別システムを構築する予定である。これは常時TCPポートを監視し、捕捉した音楽ファイルをFPGAへ転送し、データの特徴量を高速に解析することで、楽曲を識別し実時間で流通制御(例えばパケットを落とす、もしくは転送中止要求をするなど)を行なうシステムである。

文　献

[1] P. Cano, E. Batlle, H. Mayer, and H. Neuschmied, "Robust sound modeling for song detection in broadcast audio," in AES 112th International Convention, May 2002.

[2] E. Allamanche, J. Herre, O. Hellmuth, B. Frbach, and M. Cremer, "Audioid: Towards content-based identification of audio material," in 100th AES Convention, May 2001.

[3] A. Wang, "An industrial strength audio search algorithm," in 4th Int. Symposium on Music Information Retrieval (IS-MIR), Oct.2003

[4] Y. Cheng, "Music database retrieval based on spectral similarity," in 2nd Int. Sympo- sium on Music Information Retrieval (IS- MIR), Oct. 2001.

[5] P. Cano, E. Batlle, T. Kalker, and J. Haitsma, "A review of algorithms for audio fingerprinting," in International Workshop on Multimedia Signal Processing,, Dec. 2002.

[6] Jaap Haitsma , Ton Klker "A Highly Robust Audio Fingerprinting System" Proc. ISMIR 2002 3rd International Conference on Music Information Retrieval

[7] Pedro Cano , Eloi Batlle , "A Review of Algorithms for Audio Fingerprinting" Proc.IEEE International Multimedia Signal processing 2002

[8] Kazuhiro SAKAKIBARA,Yasushi INOGUCHI, "Detection of the audio watermark on FPGA", CPSY2004-80, VLD2004-114, pp. 25-30

[9] 小野 東, 電子透かしとコンテンツ保護, オーム社, 2001.

[10] 安田 浩, 安原 隆一, コンテンツ流通, アスキー, 2003.