

周波数領域のデジタルデータにおける 音声処理手法に関する検討

正木 範朗[†] 松崎 隆哲[‡]

近畿大学大学院 産業技術研究科[†] 近畿大学 産業理工学部[‡]

1. はじめに

近年、音声処理技術の高性能化に伴い雑音除去の計算量が増加しているため、ソフトウェアで雑音除去の実時間処理を行うことは困難となっている。実時間の雑音除去を実現する方法としては、高性能なコンピュータを用いて計算時間を短縮する方法とハードウェアで雑音除去回路を構築することで計算の効率化を行う方法の二通りを考えることができる。そこで、本研究ではハードウェアとして FPGA ボードを用いて実時間での雑音除去の実現を目指している。

本研究では、FPGA ボードを用いてハードウェアに適した雑音除去手法についての研究を進めている。現在は FPGA ボードに搭載されているオーディオコーデックを用いて A/D 変換及び D/A 変換を行い、フーリエ変換および音声処理を行うことで FPGA ボードを用いた音声処理の実時間処理の実現を目指している。

2. FPGA ボードを用いた音声処理(雑音除去)

本研究では、マイクロフォンを用いて FPGA ボードに音声データの入力を行い音声処理を行っていく。マイクロフォンから取り込まれた音声データは、アナログデータとして FPGA ボードに取り込まれる。しかし、FPGA ではアナログデータを取り扱うことができないため A/D 変換を行う。また、スピーカーを用いて音声処理後の音声データを出力するが、FPGA から出力される音声データはデジタルデータのため D/A 変換を行う。そこで、本研究では FPGA ボード上のオーディオコーデックを用いて音声データの A/D 変換及び D/A 変換を行う。

FPGA 上に雑音除去回路の構築を行い、オーディオコーデックで A/D 変換されたデジタルデータに対して雑音除去を行う。しかしながら、

雑音除去は周波数領域のデジタルデータに対して処理を行うが、オーディオコーデックで扱われるデジタルデータは時間領域のため、時間領域と周波数領域の相互変換を行う回路が必要となる。そのため本研究では、離散フーリエ変換回路と逆離散フーリエ変換回路を FPGA 上に構築することで時間領域と周波数領域の相互変換を行う。

3. オーディオコーデックを用いた音声データの取り扱い

本研究で用いる FPGA ボードに搭載されているオーディオコーデックの初期設定では、ADC や DAC 等のデバイスが無効となっているため A/D 変換及び D/A 変換を行うことができない。オーディオコーデックは内部に制御用のレジスタを 11 個持っており、レジスタ内のデータを更新することでデバイスを有効にすることができる。そこで、まずはオーディオコーデックのレジスタの更新を行う回路の構築を行った。

オーディオコーデックのレジスタの更新を行うためには FPGA からの制御が必要となり、I²C と SPI のどちらかの通信方式を用いて制御を行う。本研究では、仕様の明確さと信号線の少なさから I²C を用いてオーディオコーデックの制御を行った。

4. オーディオコーデックの制御回路

本研究では、FPGA 上にオーディオコーデックの制御回路を構築し、以下の手順で制御を行った。

- 手順 1 デバイスの指定
FPGA ボード上に搭載されているデバイスの中からオーディオコーデックを指定する。
- 手順 2 レジスタの指定
オーディオコーデック内のレジスタから制御内容の変更を行うレジスタを指定する。
- 手順 3 レジスタの更新
FPGA から新しいデータを送信し指定したレ

A study of digital data processing in frequency domain

[†]Noriaki Masaki, Graduate School of advanced technology
Kinki University

[‡]Takanori Matsuzaki, Faculty of Humanity-Oriented Science
and Engineering Kinki University

レジスタを更新する。

以上の手順でオーディオコーデックの制御を行ったが、FPGA からレジスタ内のデータを確認することができないため正常にレジスタ内の更新できているかが不明である。

そこで、FPGA とオーディオコーデックの通信状況を確認するために Altera 社が提供する Signal TapII を使用した。図 1 に FPGA とオーディオコーデックの通信状況を観測した波形を示す。

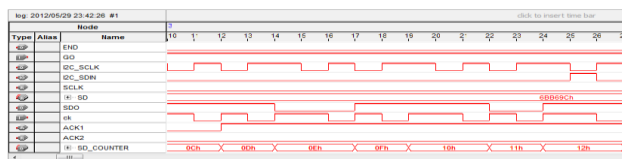


図 1: オーディオコーデックの制御波形

5. FIR フィルタ回路の設計。

オーディオコーデックの設定を変更するだけでは雑音除去などの音声処理を行うことができない。そこで、音声処理を行うための第一歩として FIR フィルタの設計を行うことにした。FIR フィルタは遅延器と乗算器、加算器から構成される。

本研究では、遅延器として FPGA 内の RAM を用い、タップ数が 192 個、サンプリング周波数が 8kHz の FIR フィルタの設計方法について検討を行った。図 2 に設計した FIR フィルタのブロック図を示す。RAM からのデータ読み出し方法としてデュアルポートモードを用いて設計の検討を行ったところ、タップ数が 192 個あるためデータをすべて読み出すためには 96 サイクル必要となった。しかし、一回の FIR フィルタの処理で行うことができるのは実際の音声データの一部であるので、一度の処理に 96 サイクルかかると実時間での音声処理は困難である。

そこで、遅延器として D フリップフロップを用いてビット幅が 16 ビットのシフトレジスタ回路を構築することで FIR フィルタの設計を行うことにした。図 3 に設計した FIR フィルタのブロック図を示す。現在、設計を行っている FIR フィルタのタップ数は 192 個で、ビット幅が 16 ビットであることから、3072 個の D フリップフロップが必要となる。D フリップフロップを用いた FIR フィルタでは一つのタップにつき一つの乗算器で計算をおこなうことができるため、処理時間の問題は解決することができると予測している。

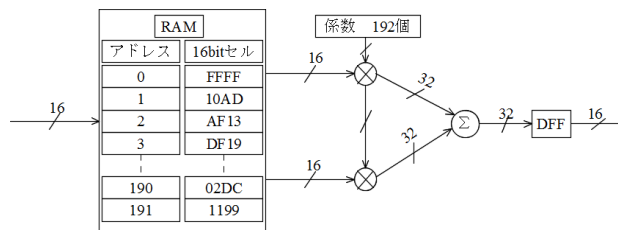


図 2: RAM を用いた FIR フィルタのブロック図

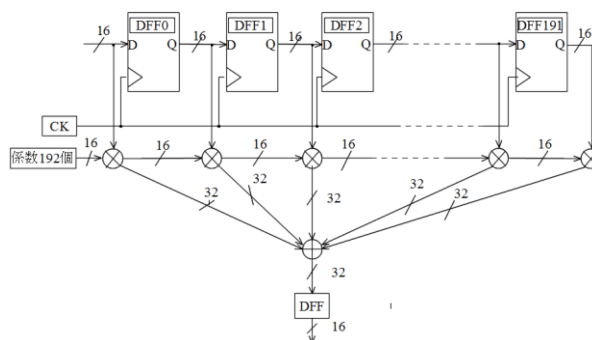


図 3: D フリップフロップを用いた FIR フィルタ

6. おわりに

本研究では、FPGA ボードを用いてハードウェア上での処理に適した雑音除去手法についての研究を進めている。本稿では、FPGA ボード上のオーディオコーデックを用いた音声データの取り扱い方法についてと FPGA 上への FIR フィルタ回路の設計方法について説明した。まずは、FPGA ボードで音声データを取り扱うために、FPGA を用いてオーディオコーデックのレジスタの更新を行った。しかし、レジスタの更新だけでは FPGA ボードで音声データを取り扱うことができないため、現在は音声処理を行うために FIR フィルタ回路の設計を行っている。

今後は、FPGA 上へ離散フーリエ変換回路及び逆離散フーリエ変換回路の構築と音声処理アルゴリズムの調査を行い FPGA 上に構築することで、オーディオコーデックを用いた音声処理システムの構築を目指す。

参考文献

[1] Altera Corporation: “Data Conversion HSMC Reference Manual”, (2008)
 [2] Texas INSTRUMENTS: “TLV320AIC23B Data Manual”, (2004)