

## 地上デジタル OFDM 受信フロントエンド部の ソフトウェアラジオ方式による実装に関する検討

鹿島 大吾<sup>†</sup> 和田 知久<sup>†</sup> 村上 修二<sup>‡</sup>

<sup>†</sup> 琉球大学理工学研究科 〒903-0129 沖縄県中頭郡西原町千原 1 番地

<sup>‡</sup> 株式会社マグナデザインネット 〒901-0152 沖縄県那覇市小禄 1831 番地 1

E-mail: <sup>†</sup> kashima@lsi.ie.u-ryukyu.ac.jp, <sup>†</sup> wada@ie.u-ryukyu.ac.jp, <sup>‡</sup> murakami@MagnaDesignNet.com

あらまし 直交周波数分割多重 OFDM(Orthogonal Frequency Division Multiplexing)変調方式はマルチパスに強く大容量の伝送が可能であり注目を集めており、高速無線ネットワークや移動体通信等の大容量デジタル通信で採用されている。本研究では地上デジタル放送における OFDM 受信機を題材にデジタル通信における信号処理を複数のプロセッサとソフトウェアで実現する所謂ソフトウェア方式の実装検討を行う。今回はフロントエンド処理であるベースバンドコンバージョン(直交復調)およびシステムサンプリングクロック誤差を補正するリサンプル演算を、オンチップマルチプロセッサによるソフトウェア処理にて実装することを目的に、その計算量・メモリ使用量・演算精度のトレードオフの検討を行ったので、その内容を報告する。

キーワード OFDM, 直交周波数分割多重, ソフトウェアラジオ, 直交復調, ダウンコンバージョン, オンチップマルチプロセッサ, リサンブラ

## Investigation on Software Radio design for Front-end Signal Processing of Terrestrial OFDM receiver

Daigo KASHIMA<sup>†</sup> Tomohisa WADA<sup>†</sup>, and Shuji MURAKAMI<sup>‡</sup>

<sup>†</sup> Graduate School of Science and Engineering, University of the Ryukyus

1 Senbaru, Nishihara, Nakagami, Okinawa, 903-0291 Japan

<sup>‡</sup> Magna Design Net, Inc. 1831-1 Oroku, Naha, Okinawa, 901-0152 Japan

E-mail: <sup>†</sup> kashima@lsi.ie.u-ryukyu.ac.jp, <sup>†</sup> wada@ie.u-ryukyu.ac.jp, <sup>‡</sup> murakami@MagnaDesignNet.com

**Abstract** The OFDM(Orthogonal Frequency Division Multiplexing) is strong to multi-path condition, which is indispensable in wireless communication, and can transmit higher bit rate. Then OFDM is getting to be utilized in many kinds of wireless digital communications, such as Terrestrial Television Broadcasting and WLANs. Since many kinds of communication protocols are proposed and have to be implemented, a reconfigurable system is strongly demanded to support many state-of-the-art communication protocols. One possible hardware platform is the on-chip multi processor system. In this paper, Software radio design based on the on-chip multi processor system is investigated for Front-end Signal processing of the OFDM receiver for ISDB-T(Integrated Services Digital Broadcasting for Terrestrial Television Broadcasting) spec. The trade-off among the number of operations, the amount of memory used, the precision of signal processing is evaluated.

**Keyword** OFDM, on-chip multiprocessor, software radio, down-conversion, resampler

### 1. はじめに

OFDM 変調方式は無線通信では不可避の電波反射によって発生するマルチパス干渉に強く、また大容量伝

送を実現できる特徴を有していることより、現在注目を集めている。OFDM 方式は高速無線ネットワークや移動体通信等の大容量デジタル通信で採用が始まって

おり、今後多種多様な OFDM 通信方式を用いた製品が登場すると考えられる。また 2003 年末に 3 都市で試験放送が開始された地上波デジタルテレビジョン放送 (ISDB-T: Integrated Services Digital Broadcasting for Terrestrial Television Broadcasting) にも採用されている。

たとえば車載関連の無線通信を考えると、上記 ISDB-T のようなデジタル放送の受信、IEEE802.11 に代表されるような WLAN、小規模な無線通信プロトコルである Bluetooth、セルラー電話関連の通信プロトコルなど多彩な通信処理に対応する必要がある。このような多種多様な通信処理を実現する一方式としてソフトウェア方式が提案されている。本方式は、デジタル通信における信号処理をプロセッサとソフトウェアを用いて実装する方法であり、ソフトウェアを入れ替えることで多様な機能を実現できるメリットがある。

本研究では、地上デジタル放送における OFDM 受信システムを題材として、そのフロントエンドにあたるベースバンドコンバージョン（直交復調）およびシステムサンプリングクロック誤差を補正するリサンプル演算部を、オンチップマルチプロセッサをソフトウェアラジオシステムのハードウェアプラットフォームとして想定し、ソフトウェア処理で実装を行う際のその計算量・メモリ使用量・演算精度のトレードオフの検討を行った。

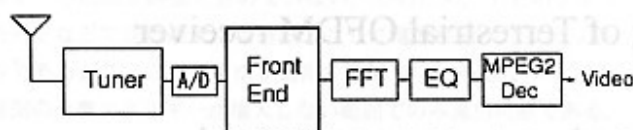


図 1 OFDM 受信システム

## 2. OFDM 受信復調フロントエンド

OFDM 受信器の構成例を図 1 に示す。アンテナで受信された電波信号はチューナにおいてあるチャンネルの周波数帯の電波信号が選択され、中間周波数帯へコンバートされる。その後、A/D 変換器によりアナログデータはデジタルデータに変換され、中間周波数帯信号はデジタル処理によりベースバンド信号に変換され I, Q からなる複素ベースバンド信号となる。その後、システムの A/D サンプリングクロック誤差を補正するリサンプル動作が行われる。OFDM では主たる復調動作が高速フーリエ変換 (FFT) であり、図 1 に示されるように FFT により復調が行われる。そして、イコライザ (EQ) により、復調出力は補正され、ビット情報に変換される。地上デジタル放送は MPEG2 のストリームを伝送するので、図のように MPEG2 デコーダを

接続することにより、ビデオ信号を得ることができる。

ここで、フロントエンド処理とはベースバンド信号を生成するコンバータ、デシメーション処理、リサンブラ処理までの FFT 前処理を示している。

### 2.1. ダウンコンバージョン

OFDM 受信機のフロントエンド部を図 2 に示す。チューナにより中間周波数帯 (IF 帯) に変換された OFDM 信号は信号処理が容易な複素ベースバンド信号に変換される。この一連の動作をダウンコンバージョン（直交復調）といい、ひとつの実現方法として図 2 に示すように、IF 信号に  $\cos(\omega t)$ ,  $\sin(\omega t)$  を乗算し、それぞれを低域通過フィルタを通すことにより、I 成分、Q 成分からなる複素ベースバンド信号を生成することができる。

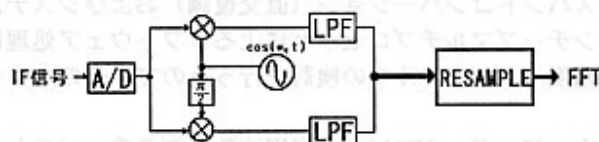


図 2 フロントエンド部の構成

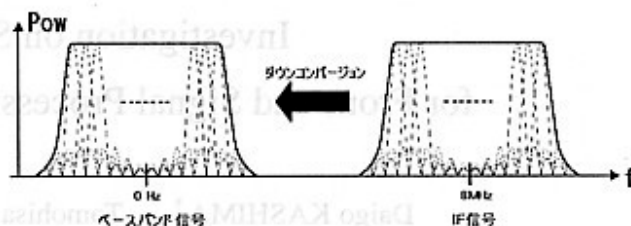


図 3 ダウンコンバージョン

### 2.2. リサンブラ、デシメーション

上記複素ベースバンド信号は IF 周波数帯信号からベースバンド信号へと周波数帯域が低下している。したがって、サンプリング定理に基づきサンプリング周波数も高周波から低周波へと落とす事が可能である。よってベースバンド OFDM 信号に対しデシメーションを施しサンプリング周波数を落とすことができる。

また通常システムの A/D サンプリング周波数は送信側のサンプリング周波数と一定の関係を保って同期する必要があり、受信システムのクロック周波数誤差を補正する必要がある。この補正を行うのがリサンブラ処理であり、システムの要求に見合ったサンプリング周波数へサンプリングレートの変換を行う。

### 3. リサンブラ

#### 3.1. 動作原理

リサンブラは前後複数のサンプリングされた値を用いてサンプリング点間の任意のタイミング値を計算する、いわゆるインターポレーション処理を行う。中間の任意のタイミングにおける補間値は、実際には複数倍にUPサンプルされた信号を生成し、その多数ある信号から適切な値を出力することにより実現される。

U倍のサンプリングレートを実現するアップサンブラは信号のサンプリング点間にU-1個の零値を挿入することで、サンプリング周波数を高くする。

図4はインターポレーションの概念図である。(簡単のためここではU=2を想定)

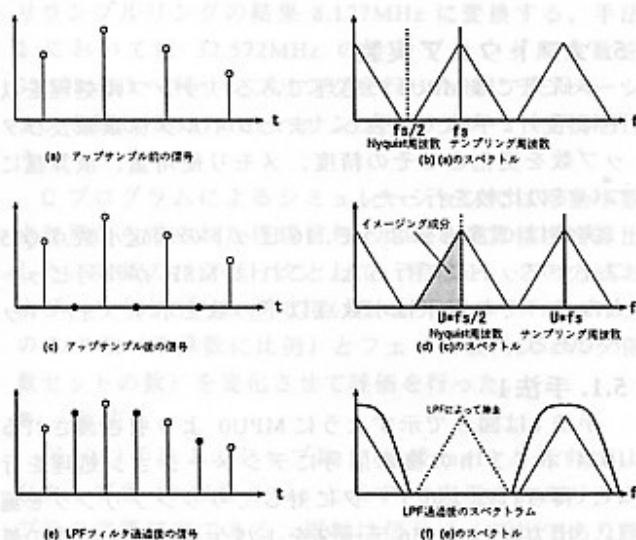


図4 インターポレーションの概念図

アップサンプリングによってサンプリングレートが増加された信号(図4(c))は零値挿入を行うだけなので処理後の信号スペクトルは変化しない。しかしナイキスト周波数もサンプリング周波数同様U倍へと高く変化しているので、それまでそれよりも低い領域にあったスペクトル成分が信号処理を行う上でイメージング成分と称される余分な成分になってしまう。(図4(d))

このイメージング成分を除去するためにアップサンプリング処理した信号をLPFに通し、あたかもU倍となったサンプリング周波数でサンプリングを行ったかのような信号を得ることが出来る。従ってオリジナル信号には存在しない任意のタイミング値を求めることができる。(図4(e))

このようなアップサンプリングの後処理用ローパスフィルタのことをインターポレーションフィルタと

言う。

#### 3.2. ポリフェーズフィルタ

前項で述べたようにインターポレーションフィルタを構成すると、U倍のサンプリング周波数で動作するフィルタが必要となる。今想定するリサンブルレートがU/Dとすると、U倍にアップサンプルした信号のうち、D個に1個の割合で出力すればよいことになる。今、U/D < 1を想定するならば、U倍にUPサンプルしたデータから多くとも1つのデータしか出力しない。そこで、実際にはU倍のサンプリング周波数で動作するフィルタは不要であり、また零挿入した後のフィルタ入力がオリジナルなサンプリング周波数ごとにしか値を持たないことを利用して、U個のフィルタに分割し、各々のフィルタはサンプリング周波数での動作を可能とするポリフェーズフィルタを使用することで動作周波数を低減することが可能である。図5はポリフェーズフィルタの概念図である。

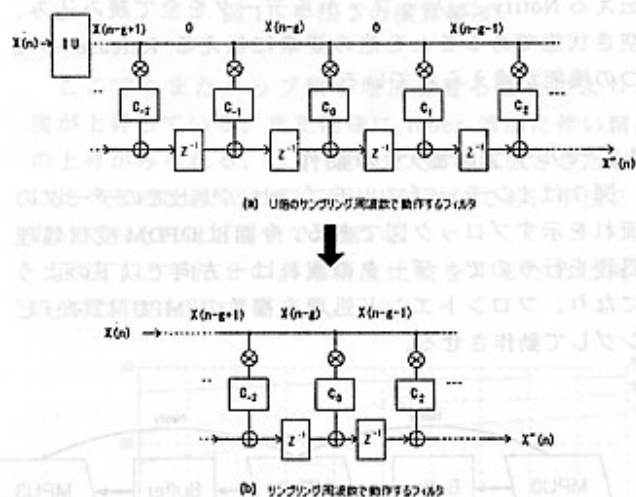


図5 ポリフェーズフィルタの概念図

図5(a)において有意のサンプル点間には零値が挿入されており、出力としては図5(b)と等価である、またフィルタ係数を変化させていくことで現時点での入力値よりリサンブル点を得る。

#### 4. オンチップマルチプロセッサシステム

オンチップマルチプロセッサシステムは、複数のプロセッサをひとつのシリコンチップ上に配置して、協調・並列動作させることで、処理の高性能化を行わせることが出来る。また同時にシステムの低コスト化および小型化を実現することが出来る。図6に、マルチプロセッサシステムの構成図を示す。

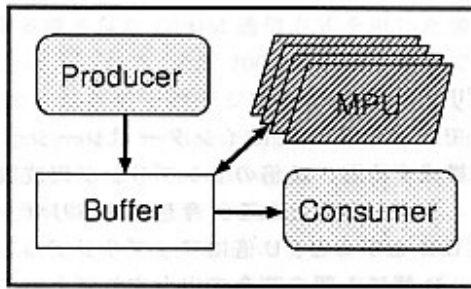


図6 マルチプロセッサシステムの構成図

Producer は外部システムからのデータ入力機能であり、Consumer は外部システムへのデータ出力機能をもつ。また MPU は複数搭載されている図6では4つのMPUを載せた場合である。これらの要素間にバッファが配置されており、要素間のデータ転送を行う。4つの要素を協調動作させて、複数のMPUによる並列実行を可能にするため、バッファをのぞいた各要素には、バッファにデータが書き込まれたことを後続の要素に伝える Notify と、バッファからデータを全て読み込み、空き状態であることを他の要素に伝える Release の2つの機能が備えられている。

#### 4.1. マルチプロセッサの動作

図7はオンチップマルチプロセッサ上でのデータの流れを示すブロック図である。今回は OFDM 受信処理を行うので、データの流は一方方向で以下のようになり、フロントエンド処理を複数のMPUにマッピングして動作させる。

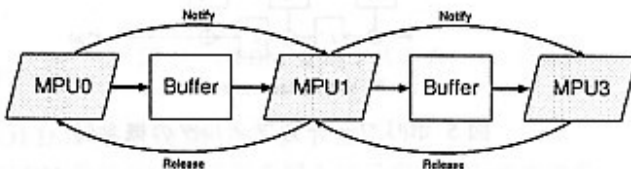


図7 オンチップマルチプロセッサ上での動作

#### ● MPU0

IF 信号に  $\cos(w_{IF}t)$ ,  $\sin(w_{IF}t)$  の信号を乗算し、それぞれに対して FIR フィルタで構成された低域通過フィルタ (LPF) 処理を行うダウンコンバージョンを実現し、MPU1 からの Release を受けると出力バッファへ出力結果、各パケットごとに 256 個 (後述の手法1の場合)、もしくは 1024 個 (手法2の場合) の複素信号データの書き込みを行う。

想定システムでは IF 周波数 (8MHz) の4倍のサンプリング周波数 (32MHz) を想定しており、上記 cos,

sin 信号は現実には 1,0,-1,0 および 0,1,0,-1 信号の繰り返しになり、乗算は不要で符号変換のための加算演算で実装可能である。

#### ● MPU1

リサンブラ処理を実現し、MPU0 からの Notify を受けると入力バッファからデータを読み込み、完了すると MPU0 へ Release を出す。処理が終了し、後続の MPU2 からの Release を受けると出力結果複素信号 256 データを出力バッファへ書き込み、終了すると MPU2 へ Notify を出し出力が終了したことを知らせる。

ここでパケットのサイズは MPU0 での 1024 ポイントから 256 ポイントに4のデシメーションが行われている。

### 5. ソフトウェア実装

本研究では MPU1 の処理であるリサンブル処理を以下に示す2手法で実装し、またフィルタ係数数及びタップ数を変化させその精度、メモリ使用量、演算量についての比較を行った。

実装は C 言語を用いて、16 ビットの固定小数点 Q15 フォーマットにて行った。これは MSB が符号ビットとなり、それ以下は小数点以下の数を示すフォーマットである。

#### 5.1. 手法1

手法1は図8で示すように MPU0 より引き渡される 1024 ポイントの複素信号にデシメーション処理を行い、得られた 256 データに対してリサンプリングを施し、出力として 256 データを1パケットとして次の処理系へ引き渡す手法である。

この手法では4のデジメーションで不要となる信号点を MPU0 で生成する必要はないので、MPU0 での演算量を削減し、同時に MPU1 への転送データのパケットサイズを 256 にすることができる。

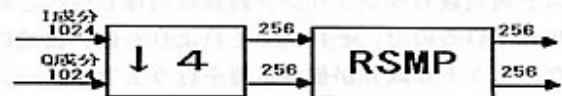


図8 手法1

#### 5.2. 手法2

手法2は図9で示すように MPU0 より引き渡される複素信号 1024 データに対しリサンプリング処理を行い、そのリサンブラ出力 1024 データにデシメーション処理を施し次の処理系へ引き渡す手法である。

この場合リサンブル後にデジメーションを行うが、実際の実装ではリサンブルとデシメーションを一体で

実装可能であり、デジメーションで廃棄されるデータポイントのリサンプリング演算量は削減可能である。

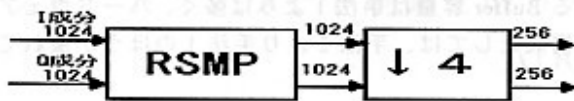


図 9 手法 2

### 5.3. シミュレーション結果

本研究では ISDB-T のスペックに従って FFT のサンプリングレートを 8.127MHz と仮定し、A/D サンプリング周波数をその 4 倍よりやや大きい 32.572MHz と想定した。手法 1 においては 32.572MHz のサンプリング周波数をその 1/4 の 8.143MHz へデジメーションし、リサンプリングの結果 8.127MHz に変換する。手法 2 においては 32.572MHz のサンプリング周波数を 32.508MHz へリサンプリングを行った後にデジメーションを施し 8.127MHz へサンプリングレート変換を行う。

C プログラムによるシミュレーションでは浮動小数点計算から得られた理想信号と比較し演算精度を算出した。実装のパラメータとして、手法 1、2 の方式および、リサンプリングで用いるポリフェーズフィルタの TAP 数（乗算数に比例）とフェイズ数（フィルタ係数セットの数）を変化させて評価を行った。

#### ● 手法 1

図 10 は手法 1 のタップ数 13(以後 tap13 と示す)およびタップ数 19(以後 19tap と示す)で実現されたリサンプリングの演算精度である。縦軸は信号ノイズ比であり横軸はフィルタ係数セットの数（以後 index と示す）を表している。

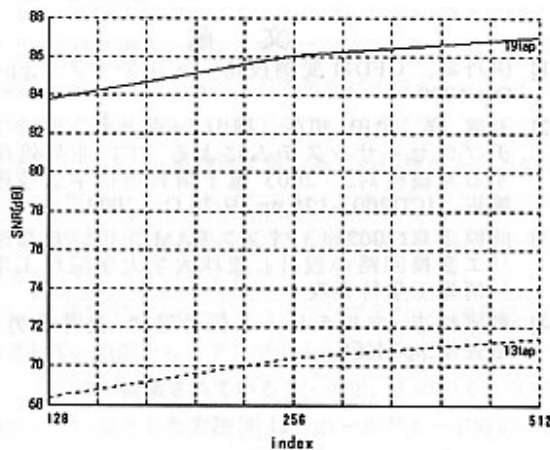


図 10 手法 1 の演算精度

13tap の約 70 dB に比べ、19tap の S/N 比は約 15 dB 高くなっており、タップ数を増加させることによ

り演算精度が格段に上昇することがわかる。また index を増加させた場合においても両者とも精度の上で改善しているが、各々約 10% あたりの改善しか見受けられない。

#### ● 手法 2

図 11 は手法 2 における演算精度結果である。

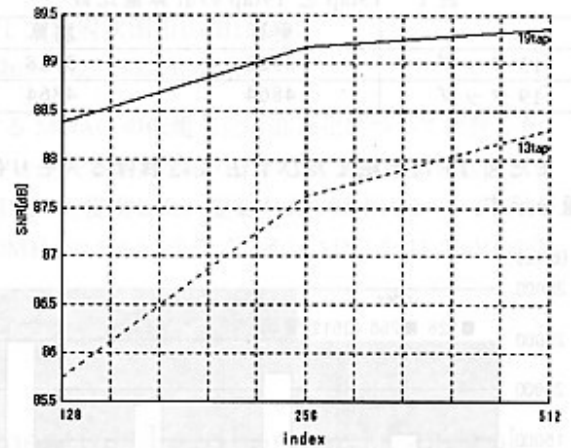


図 11 手法 2 の演算精度

ここでもまたタップ数を増加させることにより精度が上昇している。また同様に index 増加に伴い精度の上昇がみられる、しかしタップ数を増加させた場合に比べその改善率は悪く手法 1 においても、手法 2 においてもそれほど有用では無い。

手法 1 と手法 2 の演算精度比較をするために両者の演算精度を図 12 に示す。

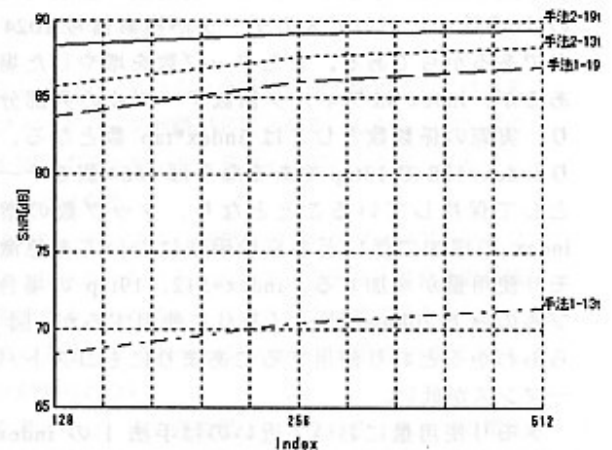


図 12 手法 1 と手法 2 の演算精度比較

13tap の場合においても、19tap の場合においても手法 2 を使用してリサンプリングを行うほうが優れており、手法 2 の有効性がわかる。しかし手法 1 においてタップ数を増やした場合精度の上昇が高く、手法 2 -13tap の精度近くまで上昇している。

### 5.3.1. メモリ使用量、演算量比較

本研究で実装したプログラムにおいて手法1と手法2では計算量自体は同じであり、表に示すように計算量はタップ数に依存する。

表1 13tapと19tapの計算量比較

	乗算	加算
13タップ	3328	3328
19タップ	4864	4864

また図13に手法1及び手法2におけるメモリ使用量を示す。

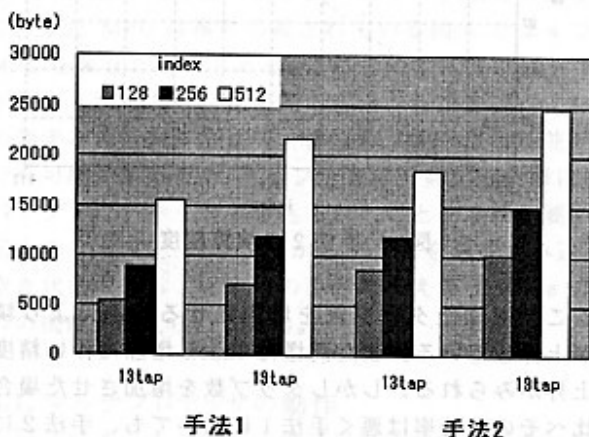


図13 メモリ使用量

手法1に比べ手法2のメモリ使用量が全般的に高いのは、手法2において入力データが複素信号1024データであるからである。またタップ数を増やした場合でも、indexはフィルタ係数テーブルの列部分であり、実際の係数数としてはindex\*tap数となる、つまりindex=128で13tapであるならば1664個をテーブルとして保持していることとなり、タップ数の増加とindexの増加に伴いどちらの手法においても急激にメモリ使用量が増加する。index=512、19tapの場合テーブルのみで20kbyte近くメモリを使用するが、図11からもわかるとおり使用するにあまりにもコストパフォーマンスが低い。

メモリ使用量において近いのは手法1のindex=256 19tapと手法2のindex=256 13tapであり精度においてもまた近い結果を得られている。しかし前者のほうが若干メモリ使用量も多く、また計算量も後者に比べて多いが精度は後者を超えないため、後者のほうが有用であると考えられる。しかしオンチップマルチプロセッサ上で動作させるためにはBufferに対しても入力値と同等のメモリ容量が必要となり、そこを加味すれば

前者のほうが有用である。よって、全般的に手法2のメモリ使用量及び、オンチップマルチプロセッサにおけるBuffer容量は手法1よりは多く、ハードウェア的な構成としては、手法2より手法1のほうが優れている。

### 6. まとめ

本研究では、オンチップマルチプロセッサシステムでの、フロントエンド部におけるリサンブラ処理に関しての検討を行った。その結果、マルチプロセッサシステムにおける構成としては、手法1を使用した場合システム全体のメモリ容量を抑えることが出来た。

また精度の上ではtap数を増加させることによって最大20%近く改善することができ、indexを変化させることによって最大10%の改善率を得られた。また手法2を使用することで最大S/N比89dB近くまで得られた。

以上のことより本研究で検討した結果においては、オンチップマルチプロセッサシステム上で動作させる最適な構造として、手法1を採用しシステム全体のメモリ容量を抑え、計算量的には増加するが手法2に近い精度を得られた19tapのリサンブラを利用が望ましいことがわかった。

今後の課題としては、更なる精度向上のアーキテクチャ考察、リアルタイム信号処理を行う際の計算時間を加味したトレードオフの検討、また図6に示すシステムの動作を実現するシミュレータの実現及び、シミュレータ上での実行速度の検討等があげられる。また、現時点ではC言語のシミュレーションレベルでの段階であるので、実機の開発が可能となれば今回の検討結果の検証および実証を行いたい。

### 文献

- [1] 伊丹誠, "OFDM 変調技術" トリケップス, pp46-51, Oct.1999
- [2] 玉置 祥, 和田 知久, 神山 一弘, "オンチップマルチプロセッサシステムによるFFT並列処理に関する基礎検討," 2003 電子情報通信学会技術研究報告, ICD2003-125, pp.19-24, Oct.2003.
- [3] 前原崇章(2002)「3バンクRAMを用いた高速フーリエ変換回路の設計」琉球大学大学院理工学研究科情報工学科専攻
- [4] 貴家仁志, マルチレート信号処理, 辻井重男(編), 昭晃堂, pp34-61.