

OFDM無線通信向き高速・低消費電力FFT回路の提案

牛木 慎祐[†] 清水 一範[†] 中村 浩一[†] 後藤 敏[†] 池永 剛[†]

[†]早稲田大学大学院情報生産システム研究科 〒808-0135 福岡県北九州市若松区ひびきの2-7

[‡]早稲田大学基幹理工学部 〒169-8555 東京都新宿区大久保3-6-1

E-mail: [†]ushiki@toki.waseda.jp

あらまし 近年、高速無線通信方式において OFDM 変調方式が注目されてきている。OFDM 無線通信においてデジタル部の主処理である FFT 回路は、搬送波数に比例して回路規模や消費電力が増大するが、高性能化のために 1024 以上の搬送波を規定している標準規格もあり、大きな課題となっている。本稿では、回路規模の増大を抑えた高速・低消費電力手法を提案する。2つの SRAM と基底 4 バタフライ演算回路を使用し、1 回のバタフライ演算に必要な 4 値をメモリの 1 ワードに格納することにより、回路規模と消費電力の増加を抑えて処理サイクル数を削減する。提案する FFT 回路を設計し、従来手法と比較して処理サイクル数を約 88%削減した。また同じスループットで比較した場合、消費電力を約 46%削減出来ることを確認した。

キーワード OFDM, FFT, 高速処理, 低消費電力, メモリアクセス

A high-throughput, low-power FFT circuit for OFDM based wireless communication systems

Shinsuke USHIKI[†] Kazunori SHIMIZU[‡] Koichi NAKAMURA[†] Satoshi GOTO[†] and

Takeshi IKENAGA[†]

[†] Graduate School of Information, Production and Systems, Waseda University

2-7 Hibikino, Wakamatsu-ku, Kitakyusyu, Fukuoka 808-0135 Japan

[‡] School of Science and Engineering, Waseda University 3-6-1 Ookubo, Shinjyuku-ku, Tokyo, 169-8555 Japan

E-mail: [†]ushiki@toki.waseda.jp

Abstract OFDM attracts attention in digital wireless communication systems. In the FFT circuit which is main processing of digital section in OFDM, a circuit scale and power consumption increase is depending on the number of the carrier waves. However, there is standards carrier waves more than 1024 for a high-performance which becomes a big problem in a hardware design. In this paper, we propose a high-throughput, low-power method which gives a good tradeoff between throughput, power dissipation and hardware complexity. Two SRAM and radix4 circuits which are necessary for 1 word memory read/write operation enables the FFT circuits to reduce the number of processing cycles. We implemented FFT circuits based on our proposed method. The proposed architecture can reduce processing cycles by about 88% compared to the conventional circuit. As a result, We can reduce power by about 46% compared to the conventional circuit in the same throughput.

Keyword OFDM, FFT, high-throughput, low-power, memory-access

1. はじめに

近年、OFDM 変調方式は、次世代の移動体通信方式の 1 つとして注目を浴びている。OFDM 変調方式は、IEEE802.11 a/g/n や IEEE802.16e (WiMAX)において標準化されており、欧州のデジタル TV(DVB-T)や ADSL において既に実用化もされている。DVB-T においては 2048 や 8192 といった極めて多い数の搬送波を使用しているが、利用可能な周波数帯域幅の増大や高性能化がこのような搬送波数の増加を促している。そのような背景の中、OFDM 変調方式において FFT はデジタル部の主要な処理回路となっているが、搬送波

数に比例して回路規模や消費電力が増大し、大きな課題となってくる。FFT は、OFDM システムの中で消費される電力のうち約 20%を占めるともいわれている [1]。UWB の規格である IEEE802.15.3a では、OFDM システムの消費電力が 200mW 以下を標準仕様としているが、その場合 FFT の消費電力は、200mW の 20%にあたる 40mW 以下に抑える必要がある。

本稿では、広い周波数帯域幅を利用する UWB-OFDM 向け FFT を主な対象とし、回路規模を抑えた高速・低消費電力な HW アーキテクチャの提案を行うと共に従来手法との比較と実装した結果を示す。

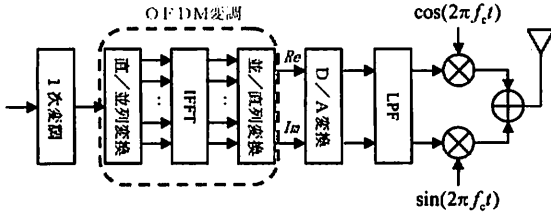


図 1: OFDM システム構成

2. OFDM 変調方式

OFDM(Orthogonal Frequency Division Multiplexing)方式とは、高速信号を分割して複数の低速信号に変換し、直交する複数の周波数帯域を使用して多重伝送する方式である。周波数帯域の有効利用とシンボル間干渉を軽減できることから無線通信技術の中で特に注目されている。図 1 に一般的な OFDM システムの構成を示す。

2.1. FFT(Fast Fourier Transform)

FFT は、DFT における計算の無駄を省く為、回転因子の周期性を上手く利用した計算技法で、次数が 2 の累乗である場合に適用できるアルゴリズムとなっている。式(1)は、N 点 FFT の計算式を表している。

$$X(k) = \sum_{n=0}^{N-1} x(n) * W_N^{nk} \quad (k=0,1,\dots,N-1) \quad (1)$$

なお、 W_N^{nk} は回転因子を表しており、式(2)のような計算式となる。

$$W_N^{nk} = \exp \left[-j \frac{2\pi}{N} * n * k \right] \quad (2)$$

2.2. バタフライ演算回路

バタフライ演算回路は、FFT を行う上での最小単位の回路構成である。一般的には Radix2(基数 2)と Radix4(基数 4)の構成が知られているが、この 2 つは 1 度に演算する値の数が異なってくる[2]。Radix2 の計算式を(3)に Radix4 の計算式を(4)に示す。

$$X(k) = \sum_{n=0}^{N-1} \left\{ x(n) + (-1)^k x\left(n + \frac{N}{2}\right) \right\} W_N^{nk} \quad (3)$$

$$X(k) = \sum_{n=0}^{N-1} \left\{ x(n) + (-j)^k x\left(n + \frac{N}{4}\right) + (-1)^k x\left(n + \frac{2N}{4}\right) + (-j)^k x\left(n + \frac{3N}{4}\right) \right\} W_N^{nk} \quad (4)$$

式(1)にて示した FFT の計算式を変形することにより、式(3)または式(4)を複数回繰返すのみで式(1)の計算結果が得られる。本稿では、式(3)または式(4)を 1 演算ステージ、繰返し回数を演算ステージ数と定義している。Radix2 と Radix4 のバタフライ演算を使用した場合を比較する。Radix4 は Radix2 に比べて複素加減算回数が約 3/2 倍、複素乗算回数が約 2/3 倍、演算

	複素加減算数	複素乗算数	演算ステージ数
Radix2	$N \log_2 N$	$\frac{N}{2} (\log_2 N - 1)$	$\log_2 N$
Radix4	$3N \log_4 N$	$\frac{3}{4} N (\log_4 N - 1)$	$\log_4 N$

表 1: Radix2 と Radix4 の演算回数

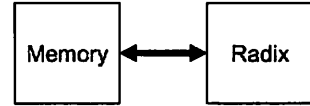


図 2: メモリベース処理

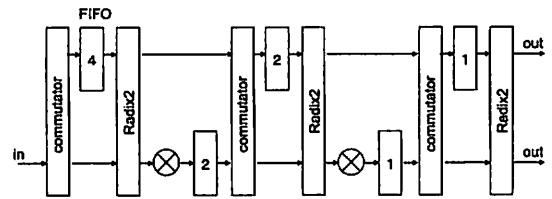


図 3: パイプライン処理

ステージ数が 1/2 倍となり、複素加減算回数は多くなるものの複素乗算回数と演算ステージ数を削減できる為、消費電力と処理サイクル数において有利な構成となっている。表 1 に N 点 FFT における Radix2 と Radix4 を使用した場合の必要な演算回数を示す。

2.3. FFT アーキテクチャ

FFT のハードウェアアーキテクチャに関しては多数の提案がなされているが、本節では基本となっている 2 つのアーキテクチャを説明する。

2.3.1. メモリベース処理

SRAM や DRAM を使用した最もシンプルな構成であり、回路規模を最も小さく出来るという特徴を持っている。N 点 FFT の場合、N ワードのメモリと 1 つのバタフライ演算回路で済む最小の構成となっている。その反面、1 つのメモリと演算器ではシリアルな処理が基本となる為、データの読み書きに必要となる時間とメモリアクセス時に要する消費電力が大きな課題となる。図 2 に回路構成を示す。

文献[5],[6]では回路規模の増加を抑制した高速処理を目的として、Radix2 演算 1 つと並列処理の為に DP-SRAM(Dual-Port SRAM)を複数使用した構成が提案されているが、DP-SRAM によるメモリ面積と消費電力の増大が課題となる。

2.3.2. パイプライン処理

主に FIFO を使用した手法であり、各演算ステージをパイプライン処理することにより高速な処理を実現できる。但し、各ステージでの演算を並列に処理する為、演算ステージ毎にバタフライ演算回路と記憶回路が必要となり、FFT の点数に比例して回路規模が増大

する。また、処理速度や必要になる回路の規模は、演算ステージ間のデータバスを単数、もしくは複数にすることにより変化する。複数のバスにした場合、バタフライ演算回路を最大限に利用することが可能となり処理速度は最大になるが、データ保持に必要となる回路規模が大きくなる。逆に単数のバスにした場合、マルチバスと比較してバタフライ演算の利用効率が落ちる為、処理速度に影響するが、データ保持に必要となる回路を若干小さくすることが可能である。図3にマルチバスにおける回路構成を示す。

3. 高速・低消費電力化手法の提案

回路をシステム LSI に実装する上で、回路規模は最も重要な要素である。2章で説明した各方式による回路規模の見積もりを図4に示す。Gate数は面積から最小サイズの2入力NANDを1Gateとして算出している。2.3.2節で示したパイプライン処理はFFTサイズの大さきによって指数関数的に回路規模が増加する。

本研究では大きなサイズのFFTを想定している為、2.3.1節の図2で示した回路規模に最も有利なメモリーブース処理を基本アーキテクチャとして採用している。1つのSP-SRAM(Single-Port SRAM)と1つのRadix4演算回路のみを使用したシンプルな構成を従来手法とし、課題となる処理速度に対する削減提案を行う。従来手法では、メモリの1ワードに1値を格納し演算に必要な値を逐次メモリにアクセスして読み書きを行っている為、処理速度において不利な構成になっている。

本提案手法では、2つのSP-SRAMと1つのRadix4演算回路を使用する。その上で処理サイクル数を削減することによる高速・低消費電力化手法を提案する。1回のRadix4演算に必要な4値をメモリの1ワードに格納することによって、演算前後の入出力値を1回のメモリアクセスで読み書きを行うことを提案の特徴としている。その為、従来手法と比較するとRead/Write処理共にメモリアクセス数を1/4に削減することが出来る。また、メモリアクセス数の削減に加えて2つのSP-SRAMを使用してRead/Writeを並列に行うことにより、処理サイクル数を約1/8に削減する。後節では、下記3点の提案手法を説明する。

- ・ 演算値のメモリ格納法
- ・ 乗算係数値のメモリ格納法
- ・ 複数值のメモリ格納に伴う演算順序

提案手法において、メモリの1ワードに4値を格納する為にはメモリのビット幅を4倍に拡張する必要がある。ビット幅を拡張すると1回のメモリアクセスにおいて消費する電力も増大する。表2はTSMC 0.18 μ mライブラリにおけるSP-SRAMとROMのビット幅とメモリアクセス時の電流量の関係を示しており、1値32ビット1024点FFTを想定している。使用するメモリアクセス数を同数とした場合、32ビット幅と4倍の128ビット幅の1回のメモリアクセスによる電流量を比較

表2: メモリビット幅と電流量の割合

		Current/1 Access		Current/Radix4	
		Read	Write	Read	Write
SP-SRAM	1024*32	1	1	4	4
	512*64	1.54	1.57	3.07	3.14
	256*128	2.80	2.84	2.80	2.84
ROM	768*32	1	-	3	-
	256*96	2.57	-	2.57	-

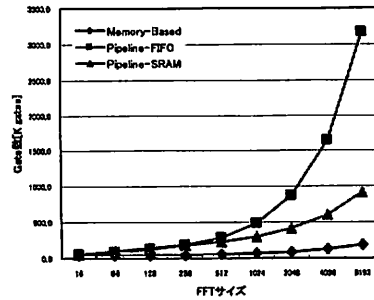


図4: 回路規模の見積もり

すると、ビット幅を4倍にした電流量の方が約2.8倍に増加する。しかし、4倍のビット幅を1回で読み書きをした場合、128ビット幅のメモリアクセス数を1/4に出来ることから4値のRead/Writeで消費される電力は4倍にはならず約2.8倍で抑えることが出来る。つまりビット幅を増やしてメモリアクセス数を削減することで電力の効率を向上させることにもつながっている。

また、乗算係数である回転因子を固定値としてROMに格納するが、SRAMを使用した場合と同様のことが言える。後節にて、別途詳細を説明する。

3.1. 演算値のメモリ格納法

本節では提案する演算値のメモリ格納法に関して説明する。提案する手法では、1回のRadix4演算に必要な4点の値を1ワードに格納し、従来4回のメモリアクセスを必要としたところを1回のメモリアクセスで1演算を行えるようにしている。

図5に1024点FFTにおける1ワード中の4点のメモリ格納例を示す。枠内の数字は処理点を表している。例えば、図5に示した2ステージ目では1回の演算に必要な『0・64・128・192』の4点を1アドレスに格納し、一方3ステージ目では1回の演算に必要な4点が変わる為、『0・16・32・48』の4点を1アドレスに格納する。つまり、演算ステージによってRadix4演算で処理する4点が変わる為、演算ステージ毎に1アドレスに格納する4点の組合せも変更している。これは読み出すメモリアドレスに格納している4点と演算後に書き込むメモリアドレスの4点が一致しない為、処理点数の2倍のメモリ容量を持つ1つのSP-SRAM、もしくは2つのSP-SRAMを使用する必要があるということも意味している。本稿では、処理速度とメモリ面

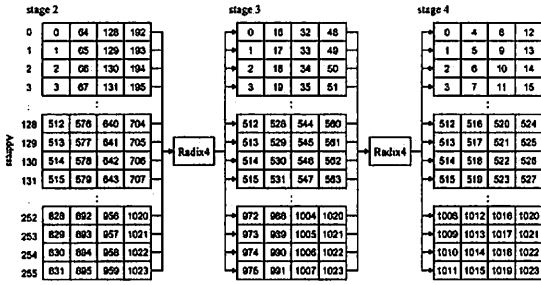


図 5：演算値のメモリ格納

表 3：係数値のメモリ格納

Address	Twiddle-Factor1	Twiddle-Factor2	Twiddle-Factor3
0	0	0	0
1	1	2	3
2	2	4	6
3	3	6	9
...
124	124	248	372
125	125	250	375
126	126	252	378
...
N/4-1	N/4-1	2(N/4-1)	3(N/4-1)

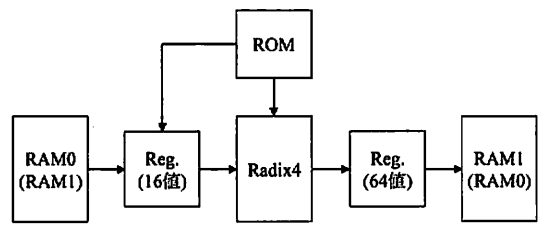


図 6：演算前後の Temporary Register

積を考慮して2つのSP-SRAMを使用している。

また、1アドレスに格納する4点の組合せを変更する為に演算前後で最低限のレジスタを設けて一時的にデータを保持している。

提案手法では1アドレスに格納する4点の組合せ変更に伴い、1ステージ目において演算前に4点FFT×4=16点分と演算後に4点FFT×16=64点分、1ステージ目以外の各ステージの演算後に4点FFT×4=16点分の値を保持する為のレジスタが最低限必要になる。但し、各演算ステージをシリアルに処理する為、演算後に必要なレジスタを演算ステージ間で共有することが出来る。従って、必要な記憶回路は演算前に16値分、演算後に64値分あれば十分ということになる。図6に演算前後に設けるレジスタ構成を示す。処理点の演算順序に関しては後節にて説明する。

3.2. 係数値のメモリ格納法

本節では提案する係数値のメモリ格納法に関して説明する。Radix4演算では、入力値に4値の演算データと乗算係数として3値の回転因子の値が必要になる。Radix4演算を使用する場合、1回の演算で使用する3つの回転因子の値は同じ組み合わせになる。表3にN点FFTで必要になる回転因子の組合せと提案するメモリの格納法を示す。表中にある Twiddle-Factor の数値は、式(2)における $n \cdot k$ の値を表しており、その値を代入した回転因子の値をメモリに格納する。

提案手法では、この性質を利用して3値を1ワードに格納し、1回のアクセスにより同時に3値をメモリから読み出しを行う。3値を1ワードに格納することにより、ROMのビット幅は3倍になるがメモリアクセス数を1/3に出来ることから3値のReadで消費される

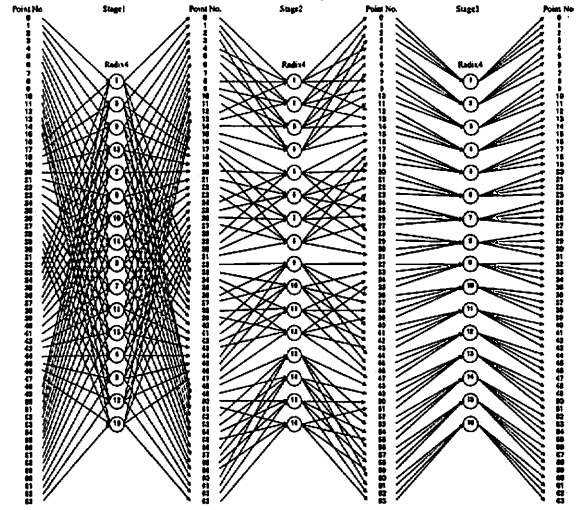


図 7：64点FFTにおける演算順序

電力は約2.6倍の増加で済む。但し、メモリ容量においては従来手法と同等であるが、メモリ面積においてはビット幅を拡張する分、若干不利になる。

また、1ステージ目を除いた演算において1アドレスに格納した回転因子の組合せは、1つの演算ステージで複数回使用される。提案手法では、S番目の演算ステージで表3に示した1アドレスの3値を4回の演算毎に 4^{S-1} 回使用する。つまり12値分の回転因子を連続して 4^{S-1} 回使用している為、前節にて説明した演算前に設けたデータ保持回路を1ステージ目以外の演算ステージでは、3値×(4回分のRadix4)=12値分の回転因子の値を保持する為に使用する。それによって、1024点FFTの場合、ROMへのメモリアクセス数を従来と比較して約1/3に削減することが出来る。

3.3. 演算順序

本節では提案するメモリ格納法に適した演算順序に関して説明する。本稿ではRadix4演算において処理する4点の規則性を考慮し、1アドレスに格納する4点の組替えによって発生する回路規模や処理サイクルの増加が極力ないような順序で演算を行っている。Radix4演算を使用したN点FFTの場合、N点をステージ毎に4分割した点の値が入力値になる。図7に64点FFTにおける演算順序を示す。丸内の数字が提案手法の演算順序を表している。演算後、次ステージで1アドレスに格納する4点の組合せを得る為に必要な4

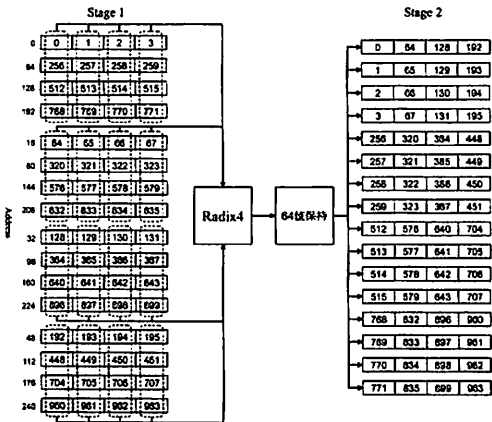


図 8：1 ステージ目における 64 値の組替え

点 FFT×4=16 点を先に演算している。

また、実際に OFDM で使用する場合は、初期値は 0 からの連続値が順に入力されるが、N 点 FFT における 1 ステージ目の Radix4 演算に必要な値は N/4 点間隔の 4 値、つまり『 $n \cdot n + N/4 \cdot n + 2N/4 \cdot n + 3N/4$ ($n=0,1,\dots,N/4-1$)』である。提案手法では、まず 0 から順に入力されるデータを連続した 4 点単位『 $0 \cdot 1 \cdot 2 \cdot 3$ 』、『 $4 \cdot 5 \cdot 6 \cdot 7$ 』…『 $N-4 \cdot N-3 \cdot N-2 \cdot N-1$ 』の値をそれぞれ 1 アドレスに格納する。次に N/16 間隔の 4 アドレスの値を 1 セットとしてメモリから読出し、1 アドレス中の値の並びにおいて同一番目となる 4 値を Radix4 演算の入力とする。次演算ステージで必要となる 4 点の組合せを得る為には、最低 4 セットの演算(16 回分の Radix4 演算)が必要になり、演算後に 64 値の組合せを変更する必要がある。その為、前章にて説明した演算前に 64 値分の記憶回路が必要となる。図 8 に 1024 点 FFT の 1 ステージ目における 64 値組替えの例を示す。

4. ハードウェア設計

前章で説明したメモリアクセス数削減による高速・低消費電力手法を適用した 1024 点の FFT 回路を Verilog-HDL を用いて設計した。本章では、ハードウェア構成と設計結果に関して説明する。表 4 に設計環境を示す

4.1. ハードウェア構成

提案する FFT 回路は、演算データを格納する 2 つの SP-SRAM(256[word]×128[bit])と回転因子データを格納する 1 つの ROM(256[word]×96[bit])、1 つの Radix4 演算回路、ビット分割・結合、Temporary Register が主な構成となっている。図 9 に設計した回路の全体構成を示す。なお、本設計では実部と虚部の値を各 16Bit とし、演算を固定小数で行っている。

動作においては、まず RAM0 に値を順に入力する。その後、RAM0 からデータを読出し、演算後データを RAM1 に格納する。次演算ステージでは、逆に RAM1 から読み出したデータを演算後に RAM0 へ格納する。

表 4：設計環境

RTL 言語	Verilog-HDL
ライブラリ	TSMC CMOS 0.18 μ m
論理シミュレータ	synopsys 社 VCS v6.2
論理合成	synopsys 社 Design Compiler v2001.08-SP2
消費電力解析	synopsys 社 Power Compiler v2001.08-SP2

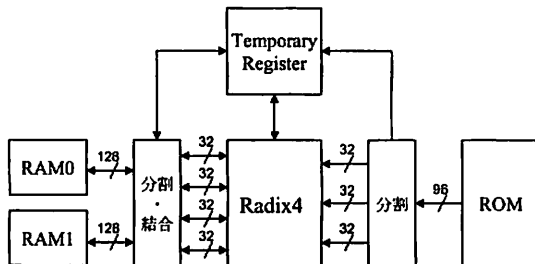


図 9：ハードウェア構成

本設計では 1024 点 FFT である為、1024 点分の Radix4 演算を 5 回繰り返す。また、一方の SRAM に書き込みを行っている間、他方の SRAM では次の計算に必要なデータの読み出しを行う。

4.2. 設計結果

本節では、3 章で示した回路を従来手法として提案手法との結果を比較する。なお、論理合成はどちらも同一の条件にて実施しており、Clock 動作周波数の制約を 100Mhz としている。また、消費電力の解析は、1024 点 FFT において約 150 μ s での処理を想定し、論理合成後の回路に対して従来手法は 100Mhz、提案手法は 12Mhz で実施している。メモリにおいては、演算データを格納する SRAM と回転因子データを格納する ROM を含めている。一方、Clock Cycle 数は、SRAM に初期値を格納した状態から FFT 演算を終了するまでを論理シミュレータによって実際に要した Cycle 数を表している。

設計結果として、回路規模は増大するが処理サイクル数を大幅に削減することが出来た。図 10 に設計結果を示す。R/L のゲート数においては、従来手法と比較すると約 1.6 倍増大している。これは、4 章にて説明した値の組合せを変更する為に設けた一時的な記憶回路による影響と、SRAM を 2 つ使用することによるアドレス制御回路の増加の影響によるものと推定する。メモリ面積の増加分も合わせて従来手法と占有面積を比較した場合、提案手法は従来手法の 1.85 倍に増大する。しかし、消費電力においては、同じスループットで従来手法と比較した場合、動作周波数を大幅に低減することができる為、約 46%削減することが出来た。3 章で説明したように元々回路規模に有利なハードウェア構成を基本としている為、今回の回路規模の増加は大きな影響ではないと考える。図 11 に提案手法による消費電力の削減効果を示す。係数値メモリで約 80%、演算値メモリで約 35%、R/L で約 33%の消費電力を削

	Conventional	Proposal
R/L Area[μm^2]	486772.85	783168
R/L ゲート数[Kgates]	48.81	78.54
メモリ数	2	3
メモリ容量[Byte]	8k	11k
メモリArea[μm^2]	392422	850182
Clock Cycle数	15370	1842
周波数[Mhz]	100	12
R/L 消費電力[mW]	9.08	6.07
メモリ 消費電力[mW]	20.11	9.85
Total 消費電力[mW]	29.19	15.91

図 10：設計結果

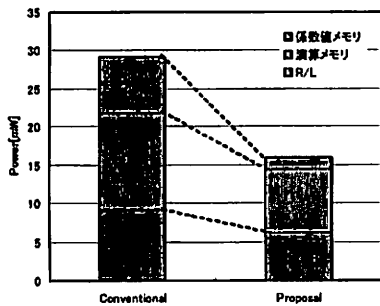


図 11：消費電力の削減割合

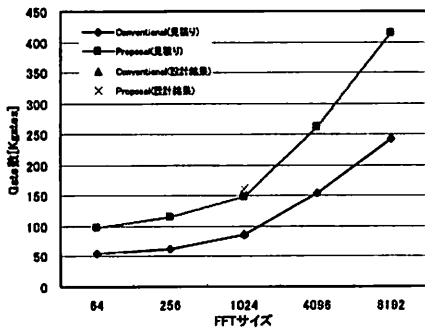


図 12：回路規模の比較

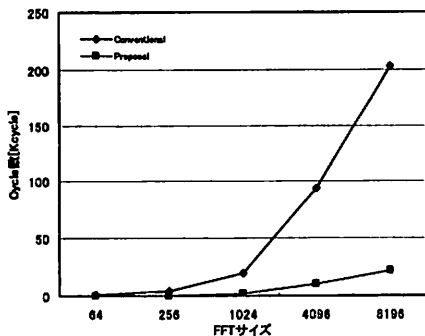


図 13：処理サイクル数の比較

減出来ている。図 12 に FFT サイズに対するメモリを含めた回路規模の見積りと設計結果、図 13 に FFT サイズに対する処理サイクル数を従来手法と提案手法において示す。処理速度においては、1024 点 FFT を従来手法の約 1/8 にあたる約 1850 Cycle で処理させることが可能となった。

5. まとめ

本研究では、UWB-OFDM での使用を目的とした大きな FFT サイズを想定し、回路規模に有利なメモリベース処理を基本とした FFT 回路において、メモリアクセス数を削減することによる高速・低消費電力化手法を提案した。SRAM の使用は、メモリアクセスに伴う消費電力の増大とデータの読み書きに要する時間が課題となる。メモリの 1 アドレスに 1 回の演算に必要な 4 値を格納することにより、回路規模と消費電力の増大を抑えて処理サイクル数を削減できた。

提案するアーキテクチャを実装した結果、従来手法と比較して処理サイクル数を約 88% 削減した。その結果、1024 点 FFT を約 150 μs で処理する場合において従来手法と比較して消費電力を約 46% 削減できた。

6. 謝辞

この研究は、独立行政法人科学技術振興機構、CREST の支援による。

文献

- [1] 坪内和夫, "ブロードバンド&低消費電力ワイヤレス通信 LSI", 第 8 回システム LSI ワークショップ, Nov.2004.
- [2] 前原崇章, "3 バンク RAM を用いた高速フーリエ変換回路の実装," 信学技法 ICD2002-167, pp.19-23, Dec.2002.
- [3] Pei-Yun Tsai, "Power-Efficient Continuous-Flow Memory-Based FFT Processor for WiMax OFDM Mode", IEEE International Symposium on Intelligent Signal Processing and Communication Systems, pp.622-625, Dec.2006.
- [4] Bass B.M., "A Low-Power, High-Performance, 1024-Point FFT Processor" IEEE J.Solid-State Circuits, vol.34, no.3, pp.380-387, Mar.1999.
- [5] Ching-Hsien Chang, "A Novel Memory-Based FFT Processor For DMT/OFDM Applications", IEEE International Conference on Acoustics, Speech, and Signal Processing, vol.4, pp.1921-1924, Mar.1999
- [6] Chao-Kai Chang., "An Efficient Memory-Based FFT Architecture", IEEE International Symposium on Circuits and Systems, vol.2, pp.129-132, May.2003