

(19) 日本国特許庁(JP)

## (12) 公表特許公報(A)

(11) 特許出願公表番号

特表2008-506191

(P2008-506191A)

(43) 公表日 平成20年2月28日(2008.2.28)

(51) Int.Cl.

G06F 17/14 (2006.01)

F 1

G06F 17/14

G06F 17/14

テーマコード(参考)

5 B 0 5 6

A

S

審査請求 未請求 予備審査請求 未請求 (全 24 頁)

(21) 出願番号 特願2007-520491 (P2007-520491)  
 (86) (22) 出願日 平成17年7月8日 (2005.7.8)  
 (85) 翻訳文提出日 平成18年12月26日 (2006.12.26)  
 (86) 國際出願番号 PCT/US2005/024063  
 (87) 國際公開番号 WO2006/014528  
 (87) 國際公開日 平成18年2月9日 (2006.2.9)  
 (31) 優先権主張番号 60/586,390  
 (32) 優先日 平成16年7月8日 (2004.7.8)  
 (33) 優先権主張国 米国(US)  
 (31) 優先権主張番号 60/586,391  
 (32) 優先日 平成16年7月8日 (2004.7.8)  
 (33) 優先権主張国 米国(US)  
 (31) 優先権主張番号 60/586,389  
 (32) 優先日 平成16年7月8日 (2004.7.8)  
 (33) 優先権主張国 米国(US)

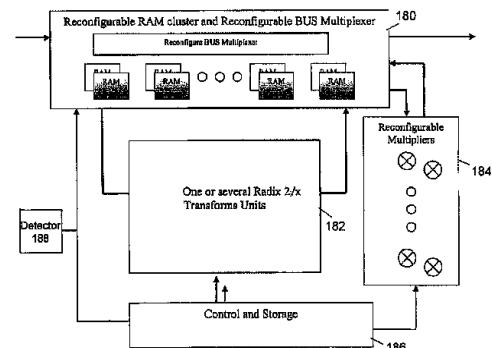
(71) 出願人 506427613  
 アソクス リミテッド  
 イスラエル ハレイ-イエフーダ, ネーヴ  
 -イラン, キリヤット ハティスクソレッ  
 ト, ルーム 220  
 (74) 代理人 100064447  
 弁理士 岡部 正夫  
 (74) 代理人 100085176  
 弁理士 加藤 伸晃  
 (74) 代理人 100094112  
 弁理士 岡部 譲  
 (74) 代理人 100096943  
 弁理士 白井 伸一  
 (74) 代理人 100101498  
 弁理士 越智 隆夫

最終頁に続く

(54) 【発明の名称】可変サイズの高速直交変換を実施する方法および機器

## (57) 【要約】

ベクトルの高速直交変換を多段階で実施する、再構成可能なアーキテクチャおよび方法であって、ベクトルのサイズはNであり、Nは変化してよく、段階の数はNの関数であり、アーキテクチャは、1つまたは複数のバタフライ・ユニットを含むように構成され配置された計算ユニット(182)と、計算ユニットの出力に結合され、変換の少なくとも1つの段階向けのバタフライ演算をすべて実施するように構成され配置された1つまたは複数の乗算器(184)を含むプロックと、各バタフライ演算を実施する計算ユニットによる使用のために、バタフライ演算の中間結果および所定の係数を格納するように構成され配置された記憶ユニット(180)であって、メモリおよび多重化アーキテクチャを(180)含む記憶ユニットとを含む。



## 【特許請求の範囲】

## 【請求項 1】

ベクトルの高速直交変換を多段階で実施する、再構成可能なアーキテクチャであって、ベクトルのサイズがNであり、Nは変化してよく、段階の数がNの関数であり、

1つまたは複数のバタフライ・ユニットを含むように構成され配置された計算ユニットと、

前記計算ユニットの出力に結合され、前記変換の少なくとも1つの段階向けの前記バタフライ演算をすべて実施するように構成され配置された1つまたは複数の乗算器を含むブロックと、

各バタフライ演算を実施する前記計算ユニットによる使用のために、前記バタフライ演算の中間結果および所定の係数を格納するように構成され配置された記憶ユニットであって、メモリおよび多重化アーキテクチャを含む前記記憶ユニットと、前記変換の前記バタフライ演算すべてを、前記段階用にただ1つの計算ユニットが必要とされるように前記一段階向けの前記計算ユニットを使って時分割するように構成され配置されたマルチブレクサ・ユニットと、

前記計算ユニットに係数を与え、前記記憶ユニット内のメモリのサイズおよび多重化アーキテクチャを制御するように構成され配置されたコントローラとを備え、

各段階用の、前記乗算器の係数、前記計算ユニットの前記係数、メモリのサイズ、および多重化アーキテクチャが、Nの値に応じて修正される、再構成可能なアーキテクチャ。

## 【請求項 2】

前記バタフライ・ユニットが、Radix<sub>2</sub>、Radix<sub>2<sup>2</sup></sub>、Radix<sub>2<sup>3</sup></sub>、Radix<sub>4</sub>、またはRadix<sub>8</sub>のアーキテクチャの1つで構成される、請求項1に記載の再構成可能なアーキテクチャ。

## 【請求項 3】

前記メモリ・レジスタが FIFOシフト・レジスタである、請求項1に記載の再構成可能なアーキテクチャ。

## 【請求項 4】

前記メモリ・レジスタの長さが、前記変換の前記段階の関数である、請求項1に記載の再構成可能なアーキテクチャ。

## 【請求項 5】

前記メモリ・レジスタの長さが、各後続段階と共に減少する、請求項1に記載の再構成可能なアーキテクチャ。

## 【請求項 6】

前記メモリ・レジスタの長さが、Nの値に応じて各段階ごとに調整される、請求項5に記載の再構成可能なアーキテクチャ。

## 【請求項 7】

前記マルチブレクサ・ユニットが、前記計算ユニットへの入力／出力ブロックを含む、請求項6に記載の再構成可能なアーキテクチャ。

## 【請求項 8】

Nが、事前定義された範囲内で変化し、事前定義された範囲全体に入力サンプル・レートでのクロッキング周波数を提供するように構成され配置されたクロック・ユニットをさらに含む、請求項1に記載の再構成可能なアーキテクチャ。

## 【請求項 9】

前記アーキテクチャが、前記事前定義された範囲M全体が、ハードウェアに対する事前定義された範囲の前記変換をマップし、前記変換がM未満のときは不必要的計算ユニットを無効にすることによって調節されるように、前記ハードウェアとして配置された多数の計算ユニットを含む、請求項8に記載の再構成可能なアーキテクチャ。

## 【請求項 10】

前記アーキテクチャが、前記事前定義された範囲M全体より小さい「m」が調節されるように、ハードウェアとして配置された多数の計算ユニットを含み、前記段階が、「m」

10

20

30

40

50

より大きい変換用の、少なくとも部分的に共有されたハードウェアである、請求項 8 に記載の再構成可能なアーキテクチャ。

【請求項 1 1】

各段階が、 $N/2$  回の計算を必要とする、請求項 1 に記載の再構成可能なアーキテクチャ。

【請求項 1 2】

複数の計算ユニットをさらに含み、1つが前記段階それぞれ用であり、前記計算ユニットが、パイプライン型アーキテクチャを提供するように実装される、請求項 1 に記載の再構成可能なアーキテクチャ。

【請求項 1 3】

複数の計算ユニットをさらに含み、1つが前記段階それぞれ用であり、前記計算ユニットが、パイプライン型、反復および並列のタイプの1つまたは複数で構成されたアーキテクチャを提供するように実装される、請求項 1 に記載の再構成可能なアーキテクチャ。

【請求項 1 4】

前記変換のフル・フレームが、 $N$  クロック周期中で実装される、請求項 1 に記載の再構成可能なアーキテクチャ。

【請求項 1 5】

前記バタフライ・ユニットが、 $Radix \times 2$  アーキテクチャを含む、請求項 1 に記載の再構成可能なアーキテクチャ。

【請求項 1 6】

前記バタフライ・ユニットが、 $Radix \times 4$  アーキテクチャを含む、請求項 1 に記載の再構成可能なアーキテクチャ。

【請求項 1 7】

前記変換のフル・フレームが、 $N/2$  クロック周期中で実施される、請求項 1 6 に記載の再構成可能なアーキテクチャ。

【請求項 1 8】

変換アクセラレータをさらに含み、前記アクセラレータが、前記計算ユニット、記憶ユニット、およびマルチプレクサ・ユニットを含み、前記アクセラレータが、前記段階すべてに対する各バタフライ演算を、反復プロセスで実施するように構成され配置された、請求項 1 に記載の再構成可能なアーキテクチャ。

【請求項 1 9】

前記記憶ユニットが、フィルタ係数を含むように構成され配置され、前記変換の最終段階の前記計算ユニットの前記乗算器が、最終段階の出力を、フィルタリングされた出力を生じるように前記フィルタ係数の1つまたは複数で乗算するように適合される、請求項 1 に記載の再構成可能なアーキテクチャ。

【請求項 2 0】

前記フィルタリングされた出力が、直交変換の逆である変換の多段階の入力に加えられ、前記段階がそれぞれ、計算ユニットを含み、前記ユニットが、パイプライン型アーキテクチャを形成する、請求項 1 9 に記載の再構成可能なアーキテクチャ。

【請求項 2 1】

前記変換が高速フーリエ変換である、請求項 1 に記載の再構成可能なアーキテクチャ。

【請求項 2 2】

前記高速フーリエ変換が、異なる  $Radix$  を含む、請求項 2 1 に記載の再構成可能なアーキテクチャ。

【請求項 2 3】

前記ベクトルが、実数ベクトルおよび複素ベクトル両方を含む、請求項 1 に記載の再構成可能なアーキテクチャ。

【請求項 2 4】

前記変換がウォルシュ直交変換を含む、請求項 1 に記載の再構成可能なアーキテクチャ。

10

20

30

40

50

**【請求項 25】**

ベクトルの高速直交変換を多段階で実施する、再構成可能なアーキテクチャを備える集積チップであって、ベクトルのサイズがNであり、Nは変化してよく、段階の数がNの関数であり、前記アーキテクチャが、

1つまたは複数のバタフライ・ユニットを含むように構成され配置された計算ユニットと、

前記計算ユニットの出力に結合され、前記変換の少なくとも1つの段階向けの前記バタフライ演算をすべて実施するように構成され配置された1つまたは複数の乗算器を含むブロックと、

各バタフライ演算を実施する前記計算ユニットによる使用のために、前記バタフライ演算の中間結果および所定の係数を格納するように構成され配置された記憶ユニットであって、メモリおよび多重化アーキテクチャを含む前記記憶ユニットと、

メモリおよび多重化アーキテクチャを含む前記記憶ユニットと、

前記変換の前記バタフライ演算すべてを、前記段階用にただ1つの計算ユニットが必要とされるように前記一段階向けの前記計算ユニットを使って時分割するように構成され配置されたマルチプレクサ・ユニットと、

前記計算ユニットに係数を与え、前記記憶ユニット内のメモリのサイズおよび多重化アーキテクチャを制御するように構成され配置されたコントローラとを備え、

各段階用の、前記乗算器の係数、前記計算ユニットの前記係数、メモリのサイズ、および多重化アーキテクチャが、Nの値に応じて修正される集積チップ。

10

20

30

40

50

**【請求項 26】**

請求項25に記載の集積チップを含む通信システム。

**【請求項 27】**

前記ベクトルのサイズを判定する検出装置をさらに備える、請求項26に記載の通信システム。

**【請求項 28】**

ベクトルの高速直交変換を多段階で実施する方法であって、ベクトルのサイズがNであり、Nは変化してよく、段階の数がNの関数であり、

計算ユニットを、1つまたは複数のバタフライ・ユニットを含むように、ブロックを、前記計算ユニットの出力に結合された1つまたは複数の乗算器を含むように構成し配列し、前記1つまたは複数のバタフライ・ユニットおよび1つまたは複数の乗算器を、前記変換の少なくとも1つの段階向けの前記バタフライ演算をすべて実施するように構成し配列すること、

各バタフライ演算を実施する前記計算ユニットによる使用のために、前記バタフライ演算の中間結果および所定の係数を記憶ユニットに格納することであって、前記記憶ユニットは、メモリおよび多重化アーキテクチャを含むこと、

前記変換の前記バタフライ演算すべてを、前記段階用にただ1つの計算ユニットが必要とされるように前記一段階向けの前記計算ユニットを使って時分割すること、ならびに

前記計算ユニットに係数を与え、前記記憶ユニット内のメモリのサイズおよび多重化アーキテクチャを制御することを含み、

各段階用の、前記乗算器の係数、前記計算ユニットの前記係数、メモリのサイズ、および多重化アーキテクチャが、Nの値に応じて修正される方法。

**【請求項 29】**

ベクトルの高速直交変換を多段階で実施する方法であって、ベクトルのサイズがNであり、Nは変化してよく、段階の数がNの関数であり、

前記計算ユニットが、前記変換の少なくとも1つの段階向けの前記バタフライ演算すべてを実施することができるよう、少なくとも1つの計算ユニットを、少なくとも1つのバタフライ・ユニットおよび前記バタフライ・ユニットの出力に結合された乗算器を含むように構成し配列することができるよう構成され配置された、再構成可能な一群のバタフライ・ユニットおよび再構成可能な1組の乗算器、ならびに前記バタフライ演算の中間

結果および各バタフライ演算の実施において使用するための所定の係数を格納するように前記計算ユニットに結合された再構成可能なメモリを使用することを含み、

各段階用の係数およびメモリのサイズが、Nの値に応じて修正される方法。

### 【請求項 30】

ベクトルの高速直交変換を多段階で実施するシステムであって、ベクトルのサイズがNであり、Nは変化してよく、段階の数がNの関数であり、

前記計算ユニットが、前記変換の少なくとも1つの段階向けの前記バタフライ演算すべてを実施することができるよう、少なくとも1つの計算ユニットを、少なくとも1つのバタフライ・ユニットおよび前記バタフライ・ユニットの出力に結合された乗算器を含むように構成し配列することができるよう構成され配置された、再構成可能な一群のバタフライ・ユニットおよび再構成可能な1組の乗算器、ならびに前記バタフライ演算の中間結果および各バタフライ演算の実施において使用するための所定の係数を格納するように前記計算ユニットに結合された再構成可能なメモリを使用することを含み、

10

各段階用の係数およびメモリのサイズが、Nの値に応じて修正されるシステム。

### 【発明の詳細な説明】

#### 【技術分野】

#### 【0001】

本出願は、以下の優先権を主張するものである。

2004年7月8日に出願した「Low-Power Reconfigurable Architecture for Simultaneous Implementation of Distinct Communication Standards」という名称の米国特許仮出願第60/586,390号(整理番号66940-016)、

20

2004年7月8日に出願した「Method and Architecture for Implementation of Reconfigurable Matrix-Vector Computations」という名称の米国特許仮出願第60/586,391号(整理番号66940-017)、

30

2004年7月8日に出願した「Method and Architecture for Implementation of Reconfigurable Orthogonal Transformations」という名称の米国特許仮出願第60/586,389号(整理番号66940-018)、

30

2004年7月8日に出願した「Method and Architecture for Implementation of Reconfigurable Trellis-Type Coding」という名称の米国特許仮出願第60/586,353号(整理番号66940-019)、

2004年8月25日に出願した「A Method And Device For On-line Reconfigurable Vitter Decoding Of Recursive And Non-recursive Systematic Convolution Codes With Varying Parameters」という名称の米国特許仮出願第60/604,258号(整理番号66940-020)、ならびに

40

2005年3月3日に出願した「Low-Power Reconfigurable Architecture For Simultaneous Implementation Of Distinct Communication Standards」という名称の米国特許出願第11/071,340号(整理番号66940-021)。

40

#### 【0002】

本開示は、たとえばFFT/IFFT(逆FFT)変換、ウォルシュ/アダマール変換など、およびこのような変換の複数のタイプの組合せを含む、変動サイズのベクトルの直交変換の実施を可能にするように、ハードウェアのオンライン再構成可能性を提供するシ

50

システムおよび方法に関する。本システムおよび方法は、このような変換を用いる通信装置において特に有用である。

【背景技術】

【0003】

一般的な直交変換は、ワイヤレス通信システムにおいて伝送される情報をエンコードする際の強力なツールを提供し、このような変換の様々なものが、情報を伝送するのに使われるプロトコルに応じて用いられる。たとえば、FFT（高速フーリエ変換）/IFFT（逆FFT）は、たとえばOFDMシステムおよびフィルタ・バンクにおけるクリティカル計算ブロックである。たとえば、N. West、D. J. Skellernの「VLSI for OFDM」（IEEE Communications Magazine、127~31頁、vol. 36、(no. 10)、1998年10月）、およびR. van Nee及びR. Prasadの「OFDM for Wireless Multimedia Communications」（Artech House Publishers、2000年）を参照されたい。10

【0004】

FFT/IFFTの魅力的な特徴は、IFFTが、FFTブロックを使って、FFTの入力および出力を結合させ、処理されたベクトルのサイズで出力を割ることによって実施することができる。したがって、同じハードウェアを、FFTおよびIFFT両方に使うことができる。FFT/IFFTを実施するいくつかの標準実装形態が公知であり、その一部は、再構成可能性をもたらす。ある標準FFT/IFFT実装形態は、FFTカーネル演算を用いている。20

【0005】

FFTカーネル演算

NポイントDFT（離散フーリエ変換）（たとえば、A. V. Oppenheim及びR. W. Schaferの「Discrete-Time Signal Processing」（Prentice Hall、New Jersey、1989）を参照されたい）のデジタル計算は、

【数1】

$$(1) \quad X[k] = \sum_{n=0}^{N-1} X[n] W_N^{nk}, \quad k \in [0, N)$$
30

であり、上式で、複素指數係数は、

【数2】

$$W_b^a = e^{-j2\pi \frac{a}{b}}$$
40

である。

【0006】

（全kに対する）DFTの直接計算は、N × N乗算およびN × (N - 1)加算を必要とする。FFTアルゴリズムは、乗算の回数を  $N \log_2 N$  に削減する、より効率的な実装である。基本的な考え方は、長さNのFFTを、長さN/2の2つのFFT構成要素に分割することであり、次いで、こうした構成要素はそれぞれ、長さN/2の2つのFFT構成要素にさらに分割され、以下同様に続く。このプロセスは、各FFT構成要素の長さが2に減じられるまで継続し、この長さは、いわゆる「バタフライ」ユニットによって直接

50

計算することができる。このようなバタフライ・ユニットのトレリスを、図1に示してある。

一般的に使われる他の2つのFFTアルゴリズムは、周波数分割(DIF)および時間分割(DIT)アルゴリズムであり、この2つは、性質が似ている。DIFアルゴリズムは、FFT中間結果が、

【数3】

$$\begin{aligned}
 X[2r] &= \sum_{n=0}^{N/2-1} x[n] W_N^{2rn} + \sum_{n=N/2}^{N-1} x[n] W_N^{2rn} \\
 &= \sum_{n=0}^{N/2-1} x[n] W_N^{2rn} + \sum_{n=0}^{N/2-1} x[n+N/2] W_N^{2r(n+N/2)} \quad r \in [0, \frac{N}{2} - 1] \\
 &= \sum_{n=0}^{N/2-1} \underbrace{(x[n] + x[n+N/2])}_{\text{バタフライ上部分岐}} W_{N/2}^n
 \end{aligned} \tag{2}$$

10

および同様に、

【数4】

$$X[2r+1] = \sum_{n=0}^{N/2-1} \underbrace{(x[n] - x[n+N/2])}_{\text{バタフライ下部分岐}} W_N^n W_{N/2}^n \tag{3}$$

20

を有する偶部および奇部に分割されるアーキテクチャの実装を示すのに用いられる。

【0007】

標準実装

標準的な従来技術の手法において、関数特有の再構成可能性を実現するためには、計算構造を分析することが第1に必要である。FFTは、バタフライ・ブロックからなる、シヤッフル交換相互連結ネットワークと見なすことができ、このネットワークは、FFTのサイズと共に変化し、したがって、最もエネルギー効率がよい完全並列実装の柔軟性のサポートを困難にする。完全並列実装において、シグナル・フロー・グラフは、ハードウェア上に直接マップすることができる。たとえば、16ポイントFFTの場合、合計で32個のバタフライ・ユニットがあり、こうしたユニットは、図2のトレリスで示すように相互連結される。概して、NポイントFFTは、 $(N/2) \log_2 N$  個のバタフライ・ユニットを必要とする。この最大並列アーキテクチャには、高性能および低電力消費の可能性があるが、特に大きいFFTサイズに対しては、コストが高い、大きいシリコン域をもっている。

【0008】

DIF FFTによって生成された出力は、ビットを逆順にされる。たとえば、

$$X[10] = X[1010_2] = Y[0101_2] = Y[5]$$

となる。

【0009】

実装が、固定小数点演算で行われるとき、スケーリングおよびオーバーフローを扱うことは、変換器の正しい挙動にとって不可欠である。FFTの各段階でのバタフライ演算は、複素加算および複素乗算両方を伴う。各複素加算は、2回の実数加算からなり、こうした加算は、入力ワード長を1ビットだけ拡張する。各複素乗算は、4回の実数乗算および

40

40

50

2回の実数加算からなる。実数乗算は、入力ワード長を2倍にする。このように、正しい挙動を確実にするために、出力ワード長が( $M + 1$ )ビットまで増加されるか、あるいは出力が、 $M$ ビットまで切り捨てられ、または丸められる必要がある。切捨てが実施される場合、出力の最上位ビットは、 $M$ ビットで記述することができる最大値まで値を切り捨てることによって破棄されるに過ぎない。丸めが実施される場合、出力が右に1ビットだけシフトされ、最下位ビットが破棄される前に、最初に「1」が正の出力に加算される。最大数および最小数( $a + b$ )は、その最下位ビットを、加算の後でゼロ(偶数)にさせるので、丸めは、加算器のオーバーフローを引き起こさない。丸めの後、出力は、 $a$ および $b$ と同じ範囲、たとえば $M$ ビット内にある。

【0010】

10

#### 列ベースの手法

列ベースのFFTアーキテクチャにおいて、計算は、相互連結が、図3のトレリスで示されるすべての段階において同一に保たれるように再構成される。バタフライへの入力は、出力が演算されるともはや必要とされないので、出力は、同じバタフライの入力に経路指定することができ、同じバタフライはしたがって、反復方式で次および後続の段階用に再利用される(インプレース計算)。その結果、ただ一列のバタフライが必要とされ、この列は、異なる計算段階によって再利用される(時分割される)。ただし、FFT係数は、段階ごとに変更される必要がある。概して、 $N$ ポイントFFTは、 $N/2$ 個のバタフライ・ユニットを必要とし、たとえば8個のバタフライが、16ポイントFFT用に必要とされる。その電力消費は、完全並列アーキテクチャに非常に近いが、より小さい区域を必要とする。再構成可能な設計にさらに変換することは、単純な反復構造が特定のサイズ向けに最適化されるので、複雑なタスクである。並列から列ベースの実装への転換は、FFTフレームを処理するより多くのクロックを必要とする。実際、並列手法は、1クロック・サイクルでのフルFFTフレームの処理を可能にするが、列手法は、反復時分割構造により、 $\log_2 N$ 個( $\text{radix} - 2$ ベースのバタフライ・アーキテクチャを用いるとき)のクロック周期を必要とする。

20

【0011】

#### 再構成可能な設計

FFTアルゴリズムを稼動するために正規のパイプライン型アーキテクチャを選ぶことによって、FFT変換の計算量の標準下方境界によって提供されるものと比較してもエネルギー・オーバーヘッドが非常に低い、再構成可能な設計を実装することが可能である。

30

【0012】

#### パイプライン型手法

正規のパイプライン型アーキテクチャでは、ただ1つのバタフライ・ユニットが各段階ごとに使われ、完全並列手法での( $N/2$ ) $\log_2 N$ および列ベースの手法での $N/2$ と比較して、合計で $\log_2 N$ の計算量となる。16ポイントFFTの長さに対するパイプライン手法の例を、図4に示してある。各段階42a、42bおよび42cの乗算器40は、ハードウェア要件同士を区別するために、バタフライ・ユニット44a、44bおよび44cとは区別される。バタフライ・ユニット44a、44b、44cおよび44dはそれぞれ、各段階ごとの $N/2$ 回のバタフライ演算の中で時分割される。バタフライ・ユニット44cを含む段階の場合、乗算器40cは「j」である。最後のバタフライ・ユニット44dの外には、乗算器は必要ない。パイプライン型ベースの実装は、列ベースの手法よりも、FFTフレームごとに、より多くのクロック周期を必要とする。というのは、パイプライン型ベースの手法は、 $N$ ( $\text{radix} - 2$ ベースのバタフライ・アーキテクチャを用いるとき)クロック周期中にフルFFTフレームを実装することができ、列手法は、反復時分割構造により、 $\log_2 N$ ( $\text{radix} - 2$ ベースのバタフライ・アーキテクチャを用いるとき)クロック周期を必要とするからである。全段階のハードウェア実装において、FFTフレームを処理するクロック数は、障害とはならない。というのは、データは、直列に1フレームずつ挿入され、フレームごとのクロック周期数は、スループットが高いまま、一定の初期遅延に変換されるからである。

40

50

## 【0013】

単一パス遅延フィードバック (SDF) 実装については、たとえば、E. H. Wald 及び A. M. Despaine の「Pipelined and parallel-pipeline FFT processors for VLSI implementation」(IEEE Trans. Comput., 414~426 頁、1984 年 5 月) を参照されたい。この実装は、バタフライ出力をフィードバック・シフト・レジスタまたは FIFO46 (そのサイズは図 4 に与えられており、この例では、レジスタの長さは、相応して 8、4、2 および 1 である) に格納することによって、メモリをより効率的に使う。1 個のデータ・ストリームが、すべての段階で乗算器を通過する。

## 【0014】

## ハイブリッド手法

ハイブリッド手法は、列およびフィードバック手法の利益を兼ね備える。この手法は、フィードバック手法の要素を使ってメモリを保存し、列段階は、より優れたハードウェア使用に使われる。4 ビット幅の列段階バタフライ・ユニットの使用は、より広い BUS 幅および適正な再構成可能乗算器の利用と組み合わせることができる。このアーキテクチャは、高い空間使用率およびアルゴリズム効率のために必要な、正確な BUS 幅をもつものにコンバートすることもできる。

## 【0015】

反復プロセスを稼動するポピュラーなアーキテクチャを、図 5 に示す。この FFT 実装は、単一のバタフライ・ユニット 50 を使用する。この単一バタフライ・ユニット設計は主に、スケジューリングおよびメモリ・アクセス方式の最適化、すなわち、繰り返し時分割された同じバタフライ・ユニットを再利用することによって段階それぞれを実装する際の、パイプライン手法の提供に注力している。Spiffie プロセッサについては、たとえば、B. M. Baas の「A Low-power, high-performance, 1024 point FFT processor」(IEEE Journal of Solid-State Circuits, 1999 年 3 月) を参照されたい。このプロセッサは、低電力消費を達成するために、RAM52 および乗算器 56 を含むキャッシュ・メモリ・アーキテクチャを用いて、FFT アルゴリズムの正規のメモリ・アクセス・パターンを活用する例である。コントローラ 54 として示されるこのプロセッサは、どの長さの FFT も実施するようにプログラミングすることができるが、RAM52 によって与えられるキャッシュ・サイズなど、特定の特徴は、特定の FFT サイズ向にのみ最適化され、この手法は、非常に低い速度で作用する。というのは、N クロック周期が、パイプライン・アルゴリズムの完全実装による FFT フレームの計算に必要とされ、一定の初期遅延をもたらすからである。このことは、再利用されるバタフライ・ユニット 50 による段階の反復時間分割により、フル・フレームは、次の FFT フレームの扱いを始めることができるようになる前に、計算される必要がある (radix-2 ベースのバタフライ・ユニットを使うとき、N クロック周期を必要とする) ことを意味する。

## 【0016】

比較的大きい radix ベースのバタフライ・ユニット、たとえば Radix-4 ベースのアーキテクチャを使うことによって、より効率的な FFT プロセッサを作ることができる。これにより、フル FFT フレームの処理に必要とされる計算クロック・サイクルが、N/2 に削減される。高度な DSP およびチップ内に実装される FFT アクセラレータのほとんどは、Radix-2 または Radix-4 FFT プロセッサに基づく。こうしたアクセラレータは、使用が限られ (FFT 変換用のみ)、非常に低い速度を使用し、高いクロック・レート設計の必要性に苦しむ。

## 【0017】

## 多重パイプライン型手法に基づくフィルタ実装

図 6 に示すような再構成可能な反復方式を用いると、効率が高いどの種類のフィルタも相關関数も実装することができる。この実装は、図 6 の 60 を見ると最もよく分かる、IFFT がその後に続く、フィルタ係数での乗算 (時間領域乗算) 用の、FFT 変換の最終

10

20

30

40

50

段階の乗算器を使うことによって遂行される。この実装は、FFT / IFFT、たとえば離散コサイン / サイン変換 (DCT および DST) のどのサブプロダクト、ならびに (等化、予測、補間および相関の計算にも用いることができる) カスケード式FFTおよびIFFTアルゴリズムを用いるフィルタリングのような、上述したアルゴリズムの組合せであるどのアルゴリズムの実装においても効率的である。

#### 【0018】

##### 異なる radix を用いる FFT

radix-2<sub>2</sub>アルゴリズムは、特に関心をもたれている。このアルゴリズムは、radix-4 および split-radix アルゴリズムそれぞれと乗算計算量が同じであり、同時に規則的な radix-2 バタフライ構造を保持する。この空間的規則性は、VLSI 実装用の他のアルゴリズムに勝る、構造上の大きな利点をもたらす。radix-2<sub>2</sub>アルゴリズムの背景にある基本的な考え方は、正規の DIF FFT アルゴリズムの 2 段階をとり、実数 / 虚数スワップおよび符号反転のみを伴う  $W^{(N/4)}_N = j$  による自明な乗算の回数を最大にすることにある。言い換えると、FFT 係数は再構成され、非自明な乗算は、すべての 2 段階においてただ 1 つの複素乗算器が必要とされるように一段階にまとめられる (全体的論理区域を削減する)。図 7 は、このような係数再構成を表すトレリスを (並列形で) 示す。すなわち、どの 2 つのバタフライ係数、  $W^i_N$  および  $W^{(i + (N/4))}_N$  に対しても、  $W^i_N$  が除かれ、次の段階に転送され、この段階は、係数 1 および

#### 【数 5】

$\tilde{j}$

を、対応する位置に残す。この係数再構成を、係数ペアすべてに対して実施した後、一段階が、非自明な乗算なしで残される。

#### 【0019】

##### ハイブリッド・パイプライン / 多重化手法

この 10 年間、いくつかのパイプライン型 FFT アーキテクチャが提案されている。シグナル・フロー・グラフの空間規則性は、パイプライン型アーキテクチャにおいて保たれるので、高度にモジュール式であり拡張が容易である。シャッフル・ネットワーク 80 は、図 8 A に示される单一バス遅延フィードバックを介して実装され、ここで、データは、单一バス中の段階 82 の間で処理され、フィードバック FIFO レジスタ 84 は、新たな入力および中間結果を格納するのに用いられる。この方式の背景にある基本的な考え方は、データを格納し、次の段階がデータを正しい順序で受け取ることができるようにスクランブルすることである。FIFO レジスタ 84 が、入力の前半部分で一杯になると、直前の結果の後半部分が、次の段階へシフトアウトされる。この間、演算要素は迂回される。入力の前半部分は、FIFO レジスタからシフトアウトされると、到着した入力の後半と共に処理要素に供給される。この間、演算要素は、2 つの出力を操作し生成しており、1 つは次の段階 82 に直接供給され、もう 1 つは、対応する FIFO レジスタにシフトインされる。乗算器 (図示せず) は、必要な場合、radix-2<sub>2</sub> または radix-2 アルゴリズムいずれかに従って、段階の間に挿入される。このような実装において使用するためのトレリスおよびデータ・パケットを、それぞれ図 8 B、8 C に示してある。

#### 【0020】

添付の図面に対して参照が行われるが、同じ参照文字指定をもつ要素は、全体を通して同じ要素を表す。

【特許文献 1】米国特許仮出願第 60 / 586,390 号

【特許文献 2】米国特許仮出願第 60 / 586,391 号

【特許文献 3】米国特許仮出願第 60 / 586,389 号

【特許文献 4】米国特許仮出願第 60 / 586,353 号

10

20

30

40

50

【特許文献5】米国特許仮出願第60/604,258号

【特許文献6】米国特許出願第11/071,340号

【非特許文献1】N. West、D. J. Skellern、「VLSI for OFDM」、IEEE Communications Magazine、127~31頁、vol. 36、(no. 10)、1998年10月

【非特許文献2】R. van Nee、R. Prasad、「OFDM for Wireless Multimedia Communications」、Artech House Publishers、2000年

【非特許文献3】A. V. Oppenheim、R. W. Schafer、「Discrete-Time Signal Processing」、Prentice Hall、New Jersey、1989

【非特許文献4】E. H. Wald、A. M. Despain、「Pipelined and parallel-pipeline FFT processors for VLSI implementation」、IEEE Trans. Comput.、414~426頁、1984年5月

【非特許文献5】B. M. Baas、「A Low-power, high-performance, 1024 point FFT processor」、IEEE Journal of Solid-State Circuits、1999年3月

【発明を実施するための最良の形態】

【0021】

以下の開示では、変動サイズを有するベクトル（実数ベクトルおよび複素ベクトル）の高速フーリエ変換（FFT）などの直交変換を実施する方法およびシステムを説明する。変換のサイズが、オンラインで決定することができ、アルゴリズムへの入力に依存する適合アルゴリズムが実装される。このような適合アルゴリズムの例は、（1）FFT、（2）逆FFT（IFFT）、（3）FFTおよびIFFT、たとえば離散コサイン／サイン変換（DCTおよびDST）の任意のサブプロダクト、（4）ウォルシュ／アダマール変換およびその任意のサブプロダクト、たとえばCDMA、DSSS、拡散／逆拡散コア・アルゴリズム、ならびに上述したアルゴリズムの任意の組合せである。本方法およびシステムは、フィルタリングおよび他の機能に対しても使うことができ、たとえば、FFTおよびIFFTアルゴリズム（このアルゴリズムは、等化、ヒルベルト変換、予測および補間ならびに相関に対しても用いることができる）をカスケードするときに実現することができる。本方法およびシステムは、FFT／IFFTおよび上述したアルゴリズムすべての、高い効率での、かつハードウェアの高速オンライン再構成による広範なパラメータでの実装を可能にする。この実装は、上述した、異なるサイズのいくつかのFFT変換またはアルゴリズムの並列または直列実装用に指定される、装置内のハードウェアの量を大幅に減少させる。

【0022】

開示する手法は、FFTベクトルの長さに適合し、それに従ってメモリのサイズを決め、たとえば、シフト・レジスタ（またはFIFO）の長さを変え、相互接続バスを必要に応じて修正し、I/Oブロックの単純な多重化を実現することによって、柔軟性を実現しやすくする簡素化された相互接続構造を提供するように、直交変換プロセッサを修正するものである。入力サンプル・レートでのクロック周波数を使って、FFTの範囲全体を、ハードウェアに直接マップし、より短い長さのFFTには不必要的ブロックを無効にすることによって、またはより長い（ただし、より低いシンボル・レートの）ケースの場合は処理段階をたたみ、ハードウェアを時分割することによって、調節することができる。このアーキテクチャは、バッファリングも直列／並列変換も必要としない。

【0023】

このアーキテクチャは、Radix-2、Radix-2<sub>2</sub>、Radix-2<sub>3</sub>、Radix-4、Radix-8、または同様の形式を使って、またはそれに従って実装することができる。radix-4（回転係数の乗数なし）は、図9に示すように行列演算として表し、

10

20

30

40

50

図10のトレリスで示すように実施することもできる。

【0024】

図11に示してある、再構成可能な  $\text{r a d i x } 2_2$  段階の実装の実施形態は、入力マルチブレクサ111と、2段階のバタフライ・ユニット110aおよび110bと、(符号反転能力をもつ)ただ1つの汎用乗算器114および1つの交差接合ブロック116を有する2つのフィードバック・メモリ112aおよび112bと、コントローラ118とを備える。ブロック116は、IFFTとFFT処理の間を切り換えるのに使われ、そうすることによって、バタフライ・ユニット110aの出力で、乗算器の必要がなくなる。実装において、メモリ112aおよび112bの使用可能メモリのサイズは、処理されるFFTの長さを調節するために、コントローラ118によって修正することができる。変換ベクトルの長さは、検出装置117によって検出し、コントローラ118によって判定することができる。さらに、各計算段階向けの、乗算器114によって使用するための係数を格納するメモリ119が提供される。

10

【0025】

図12は、16ポイントFFTの、パイプライン型  $\text{R a d i x } - 2_2$  ( $N = 16$ )に基づく実装の実施形態を示す。本実施形態では、コントローラ128は、メモリ、この場合は各段階向けのシフト・レジスタ124それぞれのサイズを設定するための入力を与える。マルチブレクサ121も、第1段階のバタフライ・ユニット122aの入力に所望の順次入力を与えるように設定される。乗算器126a、126bおよび126cは、最初の3段階それぞれの出力の所に別々に位置付けられ、最後の段階は、乗算器を必要としない。図を見ると分かるように、乗算器126aおよび126cは、虚複素「j」に結合される段階の出力を、jで乗算することによって変換する。

20

【0026】

反復プロセスを実施するアーキテクチャを組み込む代替実施形態を、図13に示す。具体的には、図13は、16ポイントFFTを処理する半パイプライン/反復  $\text{R a d i x } - 2_2$  ( $N = 16$ )に基づく実装の例を示す。本実施形態では、フィードバックを提供する乗算器130bの出力、ならびに変換プロセッサの出力を有する2つのバタフライ段階130aおよび130bのみが必要とされる。具体的には、マルチブレクサ131の出力は、バタフライ・ユニット132aの入力に与えられる。後者は、メモリ(たとえばシフト・レジスタ134a)にフィードバックを、および「j」乗算器136aに出力を提供する。「j」乗算器136aの出力は、バタフライ・ユニット132bの入力に加えられる。後者は、メモリ(たとえば、シフト・レジスタ134b)にフィードバックを、および乗算器136bに出力を提供する。出力136bは、フィードバック・パスを介して、バタフライ・ユニット132aの入力に接続される。動作の際、コントローラ138が、メモリ134のサイズを、処理段階に応じて制御する。信号ベクトルが最初に受け取られる第1の事例において、レジスタ134aおよび134bは、それぞれ「8」、「4」で設定され、信号は、2段階を介して処理される。プロセッサの出力は無効にされ、第2段階のバタフライ・ユニット132bの出力は、フィードバック・パスを介して、バタフライ・ユニット132aの入力に加えられる。次の反復中、メモリは、コントローラによって「2」、「1」に設定される。信号は次いで、第2のバタフライ・ユニット132bの出力まで連続して処理される。プロセッサの出力は次いで、有効にされ、フィードバック・パスは、プロセッサの出力が139で与えられるように無効にされる。

30

【0027】

図14は、16ポイント・フィルタの、パイプライン  $\text{R a d i x } - 2_2$  ( $N = 16$ )に基づく実装の例の実施形態を示す。やはり、メモリのサイズは、コントローラ148によって各段階ごとに設定される。フィルタ係数は、乗算器140に加えられる。このアーキテクチャは、反復もあるいはパイプライン/反復または並列の混合物でもよい。

40

【0028】

繰り返しになるが、図14のアーキテクチャは、反復あるいはパイプライン/反復または並列アーキテクチャの混合物となるように修正することができる。

50

## 【0029】

ウォルシュ / アダマール変換の有効化

行列演算としての radix - 4 変換に関する上記の考察から、アーキテクチャは、ウォルシュ拡散 / 逆拡散関数など、他の直交信号方式を処理するように容易に変えることができるところがすぐに分かる。後者は、乗算係数を、 $\pm 1$  の自明なもので単に代用することによって、既存のアーキテクチャを用いて容易に実装することができる。さらに検討すると、非自明な係数および  $-j$  で乗算される係数のみが、変更される必要があることが分かる。さらに、非自明な乗算係数は、ウォルシュ拡散 / 逆拡散関数に必要とされる自明な乗算器の実装に必要なものをすべて、すなわち FFT IFFT と、 $-j$  での乗算との間で変化するための能力を、既にもっている。ハードウェアに対する唯一の特別な要件は、コントローラ 148 を管理することにある。

## 【0030】

一例として、「Radix 4」ウォルシュ拡散 / 逆拡散バタフライ・ユニットは、以下のように、行列演算として表すことができる。

## 【数6】

$$(4) \quad \begin{pmatrix} Y(1) \\ Y(2) \\ Y(3) \\ Y(4) \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \times \begin{pmatrix} X(1) \\ X(2) \\ X(3) \\ X(4) \end{pmatrix}$$

2つの行列表現を比較することによって、2つの変換の間の関係を理解することができる。

## 【数7】

$$(5) \quad \text{Walsh} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \Leftrightarrow \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{pmatrix} \text{FFT}$$

radix - 4 変換は複素演算なので、実数ベクトルに対する 2つの独立ウォルシュ拡散 / 逆拡散プロセスを取得する。というのは、 $\pm 1$  での自明な乗数は、I と Q 信号の間を入れ替わらないからである。したがって、この特徴は、たとえば、新しい WCDMA 標準でのように、2 finger RAKE 受信機、または複素ウォルシュ拡散 / 逆拡散関数の実装用に用いることができる。また、第 2 の独立ウォルシュ拡散 / 逆拡散関数を、特別段階として使うことも、あるいは、適正な場所で、用意された I および Q によって、より大きいウォルシュ拡散 / 逆拡散用に使うこともできる（この可能性は、図 9 ~ 14 に示す再構成可能な Radix 2<sub>2</sub> アーキテクチャにおいて既に実現されている）。

## 【0031】

図 14 に提示される実装は、たとえば、 $N \log(N)$  回の演算のみを用いて実施され

10

20

30

40

50

るFFTアルゴリズムを必要とし、したがって、いくつかのコードを併せもつCDMA変調/復調のような演算、すなわち、重いデータ・ロードにとっては非常に効率的である。

#### 【0032】

ここで、複素乗算器は、擬似ランダム・シーケンスを有するウォルシュ・シーケンスを、非常に高い効率でランダム化し/非ランダム化する、周波数領域内のフィルタの実装において使うことができる（いくつかのコードを併せもつCDMA変調/復調を扱うとき、すなわち、重いデータ・ロードの場合（CDMA/WCDMA標準において見られるように）。この効率性は、変調されたデータを（コードすべてに対して）一度だけ乗算する必要があり、すべてのコードが別々に乗算されるのではないおかげで実現される。

#### 【0033】

図15は、ウォルシュ・コードをランダム化する回転乗数が、並列アーキテクチャの最初および最後で使われるときのウォルシュ拡散/逆拡散関数への radix-4段階の変換の実施形態の例のトレリスを示す。図15は具体的には、16ポイント・ウォルシュ拡散/逆拡散関数の並列 Radix-2<sub>2</sub> (N=16)に基づく実装の例を示す。

かつkお

回転乗数を使って radix-2ベースのFFTを実施すると、乗数は、「1」にのみ変えることができる。図16は、16ポイントの、ウォルシュ拡散/逆拡散シーケンスの並列 Radix-2 (N=16)に基づく実装のトレリスの例、すなわち、変調/復調中の一連の16チップのウォルシュ拡散/逆拡散シーケンスの例を示す。

#### 【0034】

やはり、複素乗算器を上で説明したように、たとえば周波数領域内のフィルタの実装に、または擬似ランダム・シーケンスを有するウォルシュ・シーケンスのランダム化/非ランダム化に使うことができる。効率性は、変調されたデータを（コードすべてに対して）一度だけ乗算する必要があり、したがって各コードが別々に乗算されるわけではないおかげで実現される。

#### 【0035】

##### 再構成可能なハイブリッド・パイプライン列の多重化手法

図17に示すように、4ビット幅の小さい radix-2<sub>2</sub>バタフライ・ユニットの「バンク」を組み合わせて、より広いBUS radix-2<sub>2</sub>を形成することができ、小さい Radixはそれぞれ、組合せ/分裂することができる、RAMの再構成可能な制御された「バンク」に接続される。BUS分裂用の再構成可能な乗算器は、上記方法に基づいて、非常に高い使用率および低電力消費である、任意の長さのIFFT/FFT/フィルタ/相関器およびウォルシュ/アダマール変換またはその任意の下位プロダクト、たとえば、CDMA DSSSコアもしくはDDS周波数フィルタも有する、再構成可能な「処理」コアを用いて実装することもでき、様々な並列/パイプライン/反復アルゴリズム・アーキテクチャ方式を含むいくつかのアルゴリズムがどの構成中でも稼動することができるとき、任意のBUS幅が必要になる。シリコンでのコアの実装は、最大クロック・レートを有するので、必要性による再構成は、任意の数の並列/パイプライン/反復アルゴリズム・アーキテクチャ方式をもたらし、各々が、いつでもアルゴリズムおよびシリコン実装資源用に、かつモデム実装のどの標準に対しても最適化され、したがって、高い使用性能を有する非常に小型の再構成アーキテクチャをもたらす。図17は、FFT/IFFTベクトルを処理する再構成可能なME-Iコアの例を示す。

#### 【0036】

本開示を要約すると、現在の手法は、簡素化された相互接続構造を用いることによる、基本FFTプロセッサの修正を含む。こうすることにより、メモリのシフト・レジスタ（または FIFO）の長さを単に変えること、バス・サイズを必要に応じて変えること、I/Oブロックのシンプル多重化によってFFTのサイズを調整する際の柔軟性が実現される。入力サンプル・レートでのクロック周波数を使って、FFTの範囲全体を、ハードウェアに直接マップし、より短い長さのFFTには不必要的ブロックを無効にすることによって、またはより長い（ただし、より遅いシンボル・レートの）ケースの場合は処理段階

10

20

30

40

50

をたたみ、ハードウェアを時分割することによって、調節することができる。このアーキテクチャは、バッファリングも直列／並列変換も必要としない。

【0037】

一例として、 $\text{R a d i x } 2_2$  アーキテクチャを用いると、(回転係数の乗数のない)  $\text{r a d i x - 4}$  は、図9に見られる行列演算としても表すことができる。対応するバタフライ構造を、図10に提示してある。したがって、上記の例に示すように、 $\text{r a d i x } 2_2$  段階の実装は、ただ1つの汎用乗算器を有する2段階のバタフライ・ユニットと、符号乗算を伴う1つの交差接合 (IFFT / FFT変化にも必要とされる) とを必要とし、こうすることにより、乗算器の必要がなくなる。対応する構造を、図11に提示してある。16ポイントFFTの $\text{R a d i x } 2_2$  実装の対応する多段階実装(図4を参照)が、図12で与えられる。同じであるが、再構成可能な反復スイッチ機構を用いて実装される変換を、図13(一段階)および図14(多段階)に提示してある。

10

【0038】

ウォルシュ／アダマール変換の有効化：FFT / IFFTに関連して以上のように説明したプロセッサおよび方法は、他の変換用のプロセッサおよび方法を実装するのに使うこともできる。行列演算としての $\text{r a d i x - 4}$  変換の上記提示から、演算をウォルシュ拡散／逆拡散関数に変えるために必要とされることは、FFT用に使われる乗数を、 $\pm 1$  の自明な乗数で代用することだけである。さらに検討すると、非自明なもの、および $-j$  で乗算されるものだけが、変更される必要があることが分かる。さらに、FFT → IFFTの間で変化させ、 $-j$  で乗算するための能力をもつ、非自明な乗数だけが、ウォルシュ拡散／逆拡散に必要とされる自明な乗数の実装に必要である。ハードウェアに対するただ1つの特別な要件は、プロセッサの動作を管理し制御するコントローラにある。

20

【0039】

「 $\text{R a d i x } 4$ 」ウォルシュ拡散／逆拡散バタフライは、下に示すように、行列演算としても表すことができる。

【数8】

$$(6) \quad \begin{pmatrix} Y(1) \\ Y(2) \\ Y(3) \\ Y(4) \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \times \begin{pmatrix} X(1) \\ X(2) \\ X(3) \\ X(4) \end{pmatrix}$$

30

2つの行列表現を比較することによって、2つの変換の間の関係を理解することができる。

【数9】

40

$$(7) \quad \text{Walsh} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \Leftrightarrow \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{pmatrix} \text{FFT}$$

$\text{r a d i x - 4}$  変換は複素演算なので、実数ベクトル拡散／逆拡散に対する2つの独立ウ

50

オルシュ拡散 / 逆拡散プロセスを得る（というのは、 $\pm 1$ での自明な乗算器は、IとQ信号の間を交換しないからである）。この側面は、新しいWCDMA標準において提供されるような、2 finger RAKE受信機、または複素ウォルシュ拡散 / 逆拡散プロセッサの実装において有用である。第2の独立ウォルシュ拡散 / 逆拡散プロセッサは、特別段階として使い、適正な場所で、用意されたIおよびQによって、より大きいウォルシュ拡散 / 逆拡散用に使うこともできる（この可能性は、再構成可能なRadix-2アーキテクチャにおいて容易に実装することができる）。

#### 【0040】

図17に提示される実装は、たとえば、 $N \log(N)$ 回の演算のみを必要とし、いくつかのコードを併せもつCDMA変調 / 復調、すなわち、重いデータ・ロードにとって非常に効率的である。

10

#### 【0041】

ここで、複素乗算器は、擬似ランダム・シーケンスを有するウォルシュ・シーケンスを、非常に高い効率でランダム化し / 非ランダム化する、周波数領域内でのフィルタのような構成の実装に使うことができる（いくつかのコードを併せもつCDMA変調 / 復調を扱うとき、すなわち、重いデータ・ロードの場合（CDMA / WCDMA標準において見られるように）。この効率は、変調されたデータを（コードすべてに対して）一度だけ乗算する必要があり、すべてのコードが別々に乗算されるのではないおかげで達成される。

#### 【0042】

図15は、ウォルシュ・コードをランダム化する回転乗数が並列アーキテクチャにおいて必要とされるとき（開始 / 最後）のウォルシュ拡散 / 逆拡散関数へのradix-4段階の変換を提示する。Radix-2ベースのFFTの例の場合、回転乗数は、「1」にのみ変更される必要がある。変調 / 復調処理のための16チップのウォルシュ拡散 / 逆拡散シーケンスの例を、図16に示す。複素乗算器を上で説明したように、たとえば周波数領域内のフィルタの実装に、または擬似ランダム・シーケンスを有するウォルシュ・シーケンスのランダム化 / 非ランダム化に使うことができる。変調されたデータを（コードすべてに対して）一度だけ乗算する必要があるおかげで、効率性が実現される。各コードは、別々に乗算される必要がない。

20

#### 【0043】

##### 再構成可能なハイブリッド・パイプライン列多重化手法

30

最後に、一般直交変換を実装する再構成可能な装置の全体アーキテクチャを、Radix-2<sub>1</sub> / xバタフライ変換の場合の図18に簡単に示す。計算ユニットは、Radix-2、Radix-2<sub>2</sub>、Radix-2<sub>3</sub>、Radix-4、Radix-8などのバタフライ・ユニットを使用して実装することができる。この装置は好ましくは、再構成可能なRAMクラスタおよび再構成可能なBUSマルチブレクサ・ブロック180、1つまたは複数のバタフライ・ユニットを備える計算ユニット182、再構成可能な乗算器ブロック184、制御および記憶ユニット186ならびに検出装置188を備える。変換の各段階で、ユニット186は、2のバタフライ・ユニット内の乗算器の係数を、変換に従って修正する（対応する係数は、値{-1, 1, j, -j}をとり得る）。ユニット182による演算の結果は、ユニット180（やはりユニット186によって制御される）のレジスタに格納される。レジスタのサイズは、段階ごとに変更される。格納されたデータの一部は、再構成可能な乗算器ブロック184に挿入され、データは、段階およびアルゴリズムに従って、制御および記憶ユニット186によって確立された係数で乗算される。乗算の結果は、ブロック180に格納される。ブロック180のマルチブレクサは、格納されたデータの多重化に使われる。各段階ごとにわずか1つのバタフライ・ユニットおよび1つのマルチブレクサを使えばよく、この1つのバタフライ・ユニットおよび乗算器は、ハードウェアを単に再構成することによって、各段階向けに再利用することができるようであらう。

40

#### 【0044】

上述した内容の具体的なアプリケーションが、参照によって組み込まれている、200

50

5年3月3日に出願した、「Low-Power Reconfigurable Architecture For Simultaneous Implementation Of Distinct Communication Standards」という名称の、我々の同時継続出願第11/071,340号(整理番号66940-021)に記載されている。図18は、この同時継続出願において記載され権利請求されているシステムのブロック図を示す。

【0045】

したがって、図19に示すように、上で述べたチップ・アーキテクチャ要件に従うようになされた集積チップの実施形態は、以下の基本的な機能構成要素を備える。

【0046】

CPU190は好ましくは、(a)装置のconfigware部、すなわち、ネット・バス192、I/Oブロック194、RAMブロック196、メガファンクション・ブロック(群)198、相互接続ブロック200、フラッシュ・メモリ・ブロック202およびクロック204を制御し、(b)メガファンクション・ブロック(群)198、ならびにバス192、I/Oブロック194、RAMブロック196、相互接続ブロック200、フラッシュ・メモリ・ブロック202およびクロック204の構成を、チップによって処理される信号のプロトコルに応じて固定するのに必要とされる、比較的小さいコンピュータ処理ユニットである。CPU190は、比較的重要でなく単純な割当てまたはタスクを計算し、メガファンクションおよびI/Oブロックを相互接続するのに使われるバスを構成するのを助けることもできる。

10

20

【0047】

ネット・バス192は、プロトコルに応じて再構成可能である。I/Oブロック194は好ましくは、チップを外界と接続する、構成可能なI/Oブロックである。そのタスクは、アプリケーション・アルゴリズムの「コンパイルされたソフトウェア」を受け取ること、および入力データを受け取り、出力処理されたデータを届けることを含む。RAM196は、好ましくは「コンパイルされたソフトウェア命令」を格納するように、かつデータをキャッシュに入れバッファに入れるように構成されたランダム・アクセス・メモリである。メガファンクション・ブロック198は好ましくは、2つ以上のアプリケーション、すなわち、アプリケーション機能の各ドメインを、一機能として並外れた効率で計算することによって処理されるプロトコルからなる比較的重要なアプリケーション機能を含むように構成される。本ケースでは、メガファンクション・ブロック198は、本明細書において説明した直交変換の1つまたは複数、あるいはそのどの組合せも含むように構成される。相互接続ブロック200は好ましくは、CPU190、I/Oブロック194、RAM196、メガファンクション・ブロック198、ならびにフラッシュ・メモリ202およびクロック・ブロック204を含む、チップの構成要素をすべて接続する再構成可能なネット・バスを含む。相互接続ブロックは、比較的重要でなく単純な割当てまたはタスクを、好ましくは余分メモリ内で実施するように構成することもできる。最後に、フラッシュ・メモリ200は好ましくは、チップがそのプログラムを介して稼動するとき、データを格納するように働く。フラッシュ・メモリは好ましくは、一度のプログラミング動作で多数の記憶場所を消去させ、またはそこに書き込ませ、そうすることによって、それを使うシステムが、異なる場所で同時に読み取り、書き込むときに、より高い実効速度で動作することができるEEPROMの形である。より少ない複素演算用には、他のタイプのメモリを使ってよいことが理解されるべきである。情報は好ましくは、チップ内に情報を維持するのに電力を必要としないやり方で情報をシリコン・チップ上に格納することによって、フラッシュ・メモリに格納される。したがって、チップへの電力を絶つことができ、電力を全く消費することなく、情報がフラッシュ・メモリに保持される。さらに、フラッシュ・メモリは、高速読み取りアクセス時間および固体衝撃抵抗を提供し、フラッシュ・メモリを、セルラー電話およびPDAのような、電池で動く装置におけるデータ記憶装置などのアプリケーションにおいて特に望ましいものにする。

30

40

【0048】

50

ここまで説明したアーキテクチャはしたがって、集積回路として実装することができる。本アーキテクチャは、ベクトル（実数ベクトルおよび複素ベクトル両方）がサイズを変化させ得る、どのタイプの直交信号方式にも適合可能と考えられる。このような直交信号方式は、FFT変換、逆FFT変換（IFFT）または離散コサイン／サイン変換（DCTおよびDST）のようなその任意のサブプロダクト、ウォルシュ／アダマール変換またはCDMA DSSS拡散／逆拡散のようなその任意のサブプロダクト、こうしたアルゴリズムの2つ以上の組合せである任意のアルゴリズム、ならびに、等化、ヒルベルト変換、予測、補間、相関などにも使うことができる、このような他の機能、たとえば、FFTおよびIFFT変換の連結を用いることによるフィルタリングを含み得るが、それに制約されない。

10

## 【0049】

本明細書で開示した本開示のアーキテクチャ、およびその全要素は、添付の特許請求の範囲の少なくとも1つの範囲内に含まれる。ここで開示したチップ・アーキテクチャのどの要素も、権利放棄されることも意味しているのではなく、必ずしも特許請求の範囲の解釈を制限することも意図していない。

## 【図面の簡単な説明】

## 【0050】

【図1】FFTバタフライ演算トレリスを示す図である。

【図2】周波数分割16ポイントFFTトレリスを示す図である。

20

【図3】列ベースの16ポイントFFTトレリスを示す図である。

【図4】パイプラインベースのRadix-2 ( $N = 16$ ) 16ポイントFFTを実装するアーキテクチャを示すブロック図である。

【図5】簡易Radix-2 FFTプロセッサを実装するアーキテクチャを示すブロック図である。

【図6】パイプラインベースのRadix-2 ( $N = 16$ ) 16ポイント・フィルタのアーキテクチャを示すブロック図である。

【図7】係数の再構成を介した乗算除去技法のトレリスを示す図である。

【図8】シャッフル交換相互接続変換器のパイプライン型実装を示すトレリス、ブロック図およびパケット図である。

【図9】本開示の方法およびシステムの一態様による、radix-4バタフライ・アーキテクチャで使用するための行列演算を示す図である。

30

【図10】本開示の方法およびシステムの一態様によるradix-2<sub>2</sub>段階トレリスを示す図である。

【図11】本開示の方法およびシステムの一態様による、再構成可能なRadix-2<sub>2</sub>段階バタフライ配置のアーキテクチャを示すブロック図である。

【図12】本開示の方法およびシステムの一態様による、パイプラインベースのRadix-2 ( $N = 16$ ) 16ポイント・フィルタを示す図である。

【図13】本開示の方法およびシステムの一態様による、16ポイントFFTの半パイプライン／反復Radix-2<sub>2</sub> ( $N = 16$ ) ベースの実装を示す図である。

40

【図14】本開示の方法およびシステムの一態様による、16ポイント・フィルタのパイプライン型Radix-2<sub>2</sub> ( $N = 16$ ) ベースの実装を示す図である。

【図15】本開示の方法およびシステムの一態様による、16ポイント・ウォルシュ拡散／逆拡散関数の並列Radix-2<sub>2</sub> ( $N = 16$ ) ベースの実装のトレリスを示す図である。

【図16】本開示の方法およびシステムの一態様による、16ポイント・ウォルシュ拡散／逆拡散関数の並列Radix-2 ( $N = 16$ ) ベースの実装のトレリスを示す図である。

【図17】本開示の方法およびシステムの一態様による、再構成可能なMF-Iコア・プロセッサを実現するアーキテクチャを示すブロック図である。

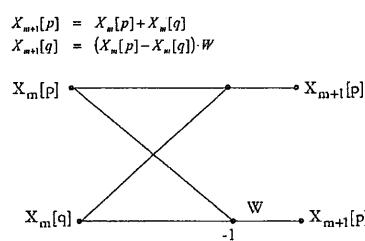
【図18】本開示の方法およびシステムの一態様による、再構成可能なMF-Iコア・プ

50

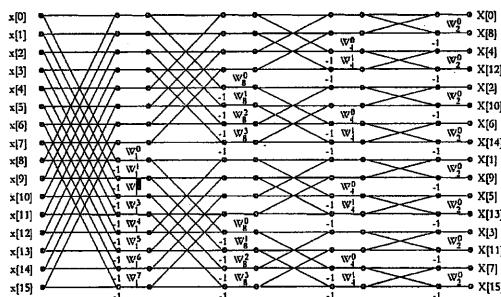
口セッサを実現するアーキテクチャを示すブロック図である。

【図19】本明細書に記載されるいづれのタイプの変換器も備えるように構成された通信システムを示すブロック図である。

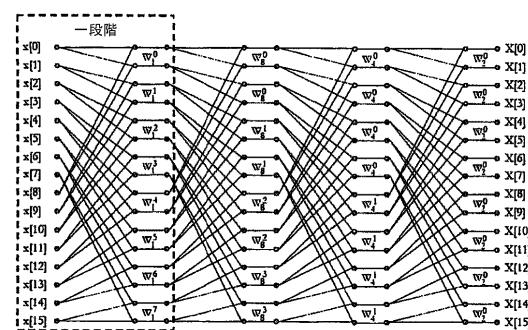
【図1】



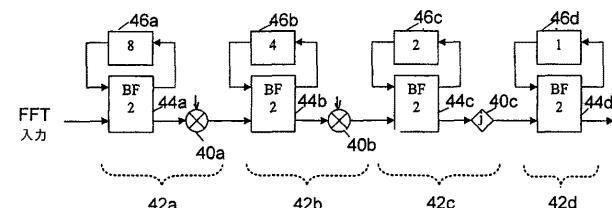
【図2】



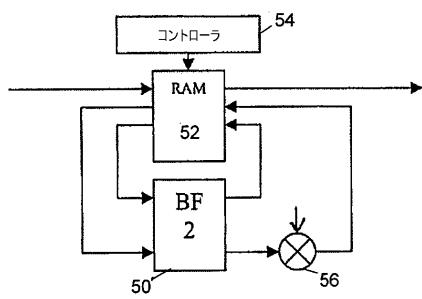
【図3】



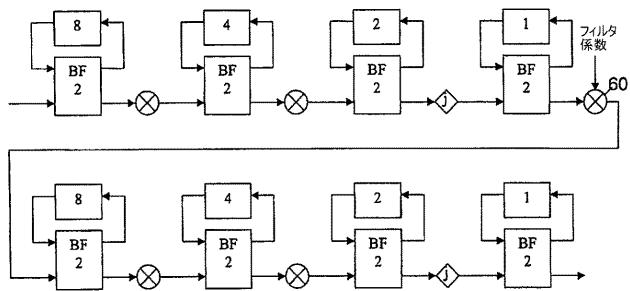
【図4】



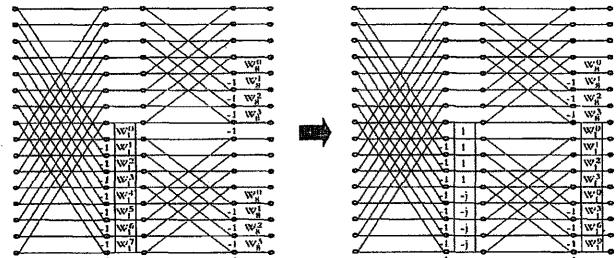
【 四 5 】



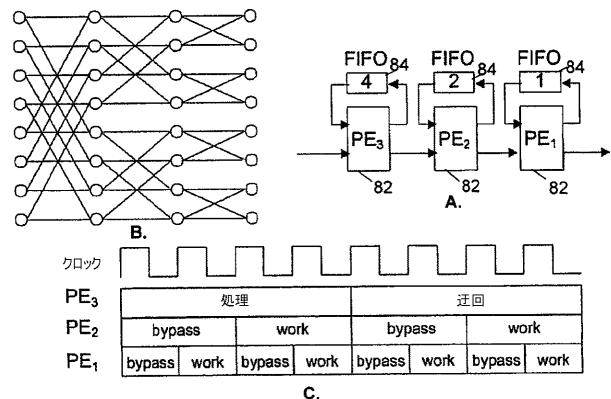
【 四 6 】



【 図 7 】



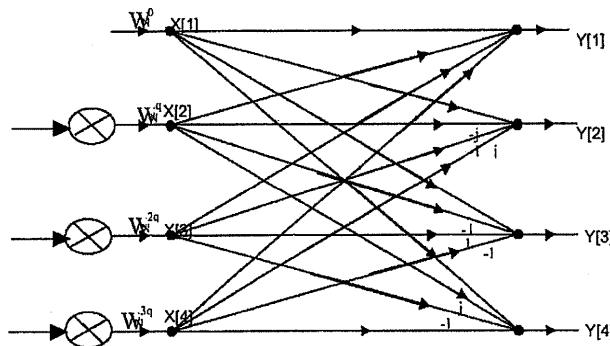
【 図 8 】



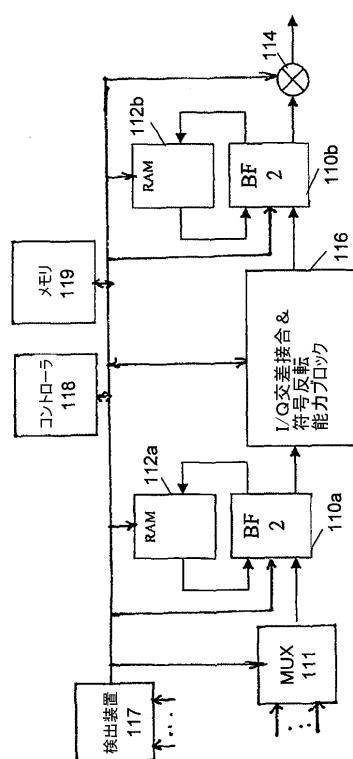
【 9 】

$$\left[ \begin{array}{c} Y(1) \\ Y(2) \\ Y(3) \\ Y(4) \end{array} \right] = \left[ \begin{array}{cccc} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{array} \right] X \left[ \begin{array}{c} X(1) \\ X(2) \\ X(3) \\ X(4) \end{array} \right]$$

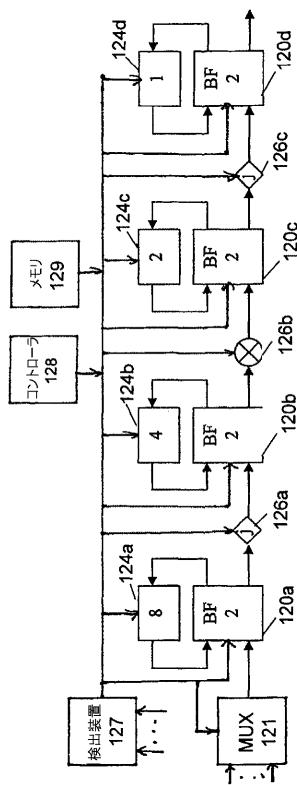
【 図 1 0 】



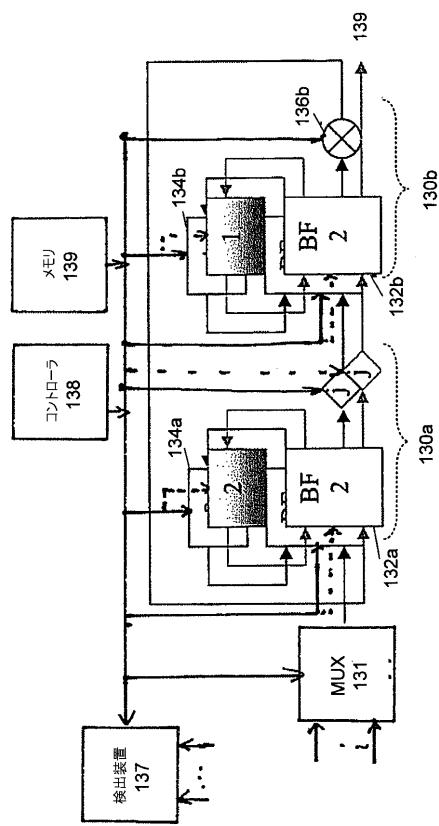
【 図 1 1 】



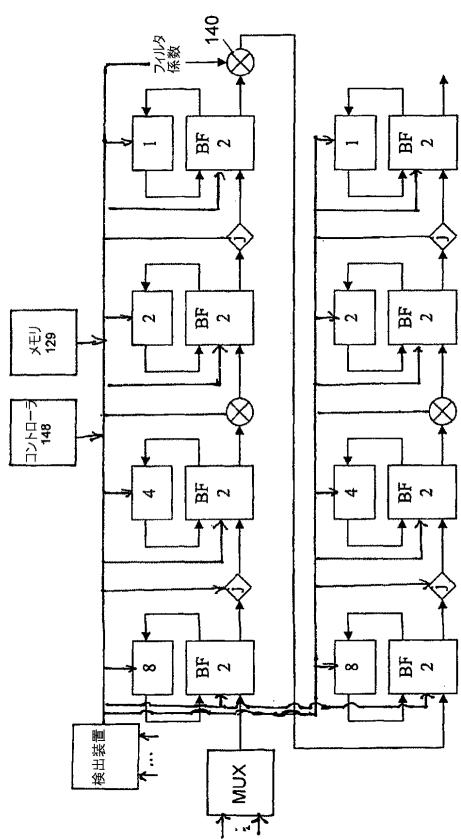
【 図 1 2 】



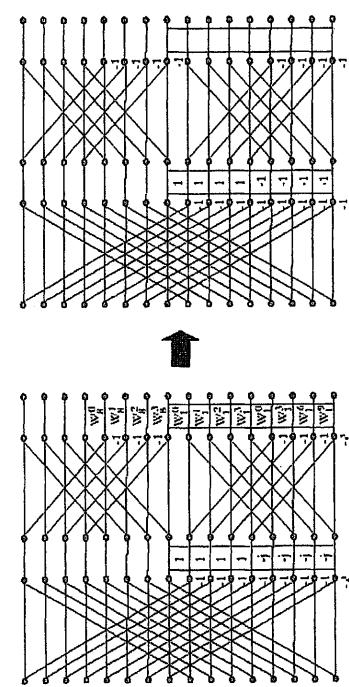
【 図 1 3 】



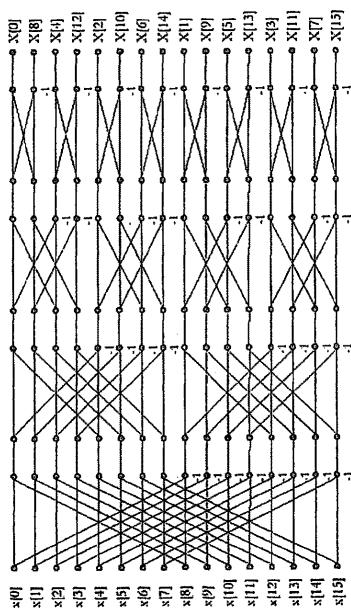
【 図 1 4 】



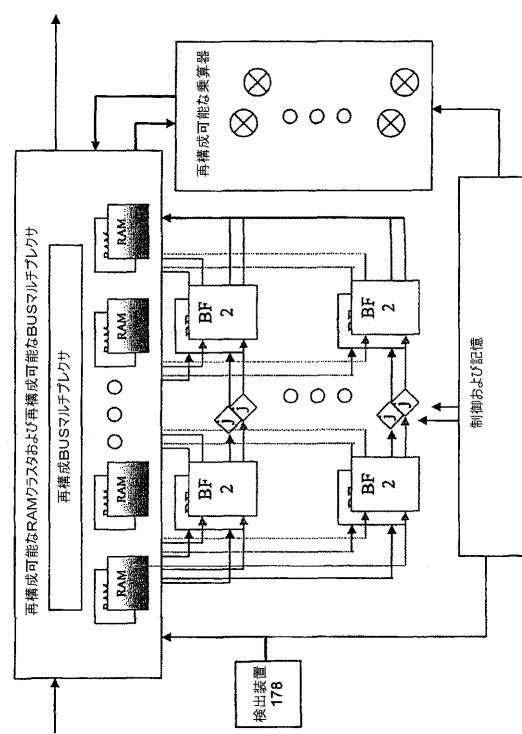
【 図 1 5 】



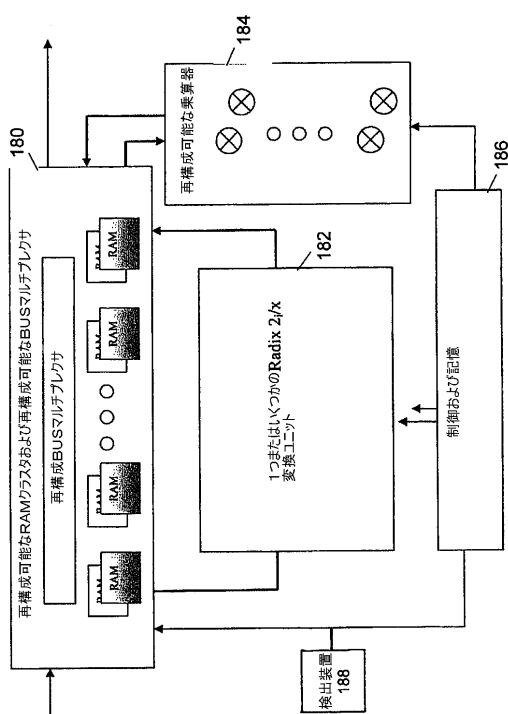
【図 16】



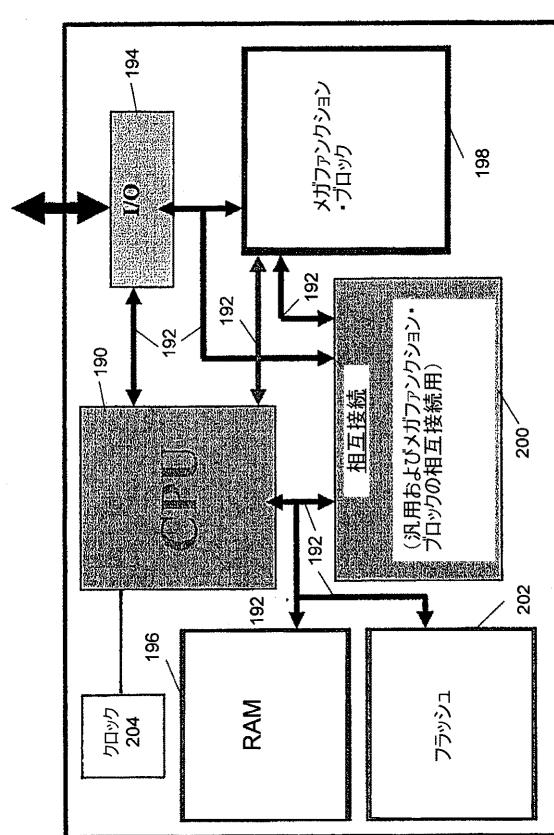
【図 17】



【図 18】



【図 19】



## 【国際調査報告】

INTERNATIONAL SEARCH REPORT		International application No. PCT/US05/24063															
<b>A. CLASSIFICATION OF SUBJECT MATTER</b> IPC(7): G06F 17/14 ; US CL: 708/400; 712/2																	
According to International Patent Classification (IPC) or to both national classification and IPC																	
<b>B. FIELDS SEARCHED</b> Minimum documentation searched (classification system followed by classification symbols) US CL: 708/400, 401, 404, 407, 408, 409, 410; 712/1, 2, 7, 11, 15, 16; 370/208																	
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched																	
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) USPTO Patent Full-Text and Full-Page Image Databases; PatentCafe ICO patent database collections; Google																	
<b>C. DOCUMENTS CONSIDERED TO BE RELEVANT</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 2px;">Category*</th> <th style="text-align: left; padding: 2px;">Citation of document, with indication, where appropriate, of the relevant passages</th> <th style="text-align: left; padding: 2px;">Relevant to claim No.</th> </tr> </thead> <tbody> <tr> <td style="padding: 2px;">A</td> <td style="padding: 2px;">US 6,237,012 B1 (OHGOSE) 22 May 2001 (22.05.2001); see entire document, particularly rearrangement means (802), column 12 lines 53-65.</td> <td style="padding: 2px;">1-30</td> </tr> <tr> <td style="padding: 2px;">A</td> <td style="padding: 2px;">US 6,061,705 A (HELLBERG) 09 May 2000 (09.05.2000); see entire document, particularly figures 8 and 9A-9C and claims 13, 40 and 43.</td> <td style="padding: 2px;">1-30</td> </tr> <tr> <td style="padding: 2px;">A</td> <td style="padding: 2px;">US 2003/0055861 A1 (LAI et al.) 20 March 2003 (20.03.2003); Figures 3D and 5; claim 33.</td> <td style="padding: 2px;">1-30</td> </tr> <tr> <td style="padding: 2px;">A</td> <td style="padding: 2px;">US 6,735,167 B1 (NAWA et al.) 11 May 2004 (11.05.2004); Figures 2 and 5, column 12 lines 28-38.</td> <td style="padding: 2px;">1-30</td> </tr> </tbody> </table>			Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.	A	US 6,237,012 B1 (OHGOSE) 22 May 2001 (22.05.2001); see entire document, particularly rearrangement means (802), column 12 lines 53-65.	1-30	A	US 6,061,705 A (HELLBERG) 09 May 2000 (09.05.2000); see entire document, particularly figures 8 and 9A-9C and claims 13, 40 and 43.	1-30	A	US 2003/0055861 A1 (LAI et al.) 20 March 2003 (20.03.2003); Figures 3D and 5; claim 33.	1-30	A	US 6,735,167 B1 (NAWA et al.) 11 May 2004 (11.05.2004); Figures 2 and 5, column 12 lines 28-38.	1-30
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.															
A	US 6,237,012 B1 (OHGOSE) 22 May 2001 (22.05.2001); see entire document, particularly rearrangement means (802), column 12 lines 53-65.	1-30															
A	US 6,061,705 A (HELLBERG) 09 May 2000 (09.05.2000); see entire document, particularly figures 8 and 9A-9C and claims 13, 40 and 43.	1-30															
A	US 2003/0055861 A1 (LAI et al.) 20 March 2003 (20.03.2003); Figures 3D and 5; claim 33.	1-30															
A	US 6,735,167 B1 (NAWA et al.) 11 May 2004 (11.05.2004); Figures 2 and 5, column 12 lines 28-38.	1-30															
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input type="checkbox"/> See patent family annex.																	
* Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed																	
Date of the actual completion of the international search 08 November 2005 (08.11.2005)		Date of mailing of the international search report <b>28 DEC 2005</b>															
Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-3201		Authorized officer: Blaine R. Copenheaver Telephone No. 571-272-7774															

---

フロントページの続き

(31) 優先権主張番号 60/586,353  
(32) 優先日 平成16年7月8日(2004.7.8)  
(33) 優先権主張国 米国(US)  
(31) 優先権主張番号 60/604,258  
(32) 優先日 平成16年8月25日(2004.8.25)  
(33) 優先権主張国 米国(US)  
(31) 優先権主張番号 11/071,340  
(32) 優先日 平成17年3月3日(2005.3.3)  
(33) 優先権主張国 米国(US)

(81) 指定国 AP(BW,GH,GM,KE,LS,MW,MZ,NA,SD,SL,SZ,TZ,UG,ZM,ZW),EA(AM,AZ,BY,KG,KZ,MD,RU,TJ,TM),EP(AT,BE,BG,CH,CY,CZ,DE,DK,EE,ES,FI,FR,GB,GR,HU,IE,IS,IT,LT,LU,LV,MC,NL,PL,PT,RO,SE,SI,SK,TR),OA(BF,BJ,CF,CG,CI,CM,GA,GN,GQ,GW,ML,MR,NE,SN,TD,TG),AE,AG,AL,AM,AT,AU,AZ,BA,BB,BG,BR,BW,BY,BZ,CA,CH,CN,CO,CR,CU,CZ,DE,DK,DM,DZ,EC,EE,EG,ES,FI,GB,GD,GE,GH,GM,HR,HU,ID,IL,IN,IS,JP,KE,KG,KM,KP,KR,KZ,LC,LK,LR,L,S,LT,LU,LV,MA,MD,MG,MK,MN,MW,MZ,NA,NG,NI,NO,NZ,OM,PG,PH,PL,PT,RO,RU,SC,SD,SE,SG,SK,SL,SM,SY,TJ,TM,TN,TR,TT,TZ,UA,UG,US,UZ,VC,VN,YU,ZA,ZM,ZW

(74) 代理人 100104352  
弁理士 朝日 伸光  
(74) 代理人 100128657  
弁理士 三山 勝巳  
(72) 発明者 ソロモン, ドロン  
イスラエル ホロン, ハタヴス ストリート 4  
(72) 発明者 ギヤロン, ギラド  
イスラエル エルサレム, ヘルツル ストリート 97  
F ターム(参考) 5B056 AA01 BB13 BB17 GG05