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

2004－ＳＬＤＭ－113　(10)
2004／1／22

N次元高速アダマール変換アルゴリズムの提案と動的再構成可能デバイスへの実装

高橋 宏章　　北道 淳司　　黒田 研一

会津大学コンピュータ理工学研究科

あらまし　ディジタル信号の直交変換処理の一つである高速アダマール変換のバタフライ演算の並列度を動的に $1/2^n$ にして処理可能なアーキテクチャを提案する。また、提案アーキテクチャを動的再構成可能デバイス上で実装する方法を述べる。提案するアーキテクチャは、処理次元、信号数および並列度を動的に変更することが可能であり、処理システムへの動的な要求変更に柔軟に対応できる。提案アルゴリズムを並列度 1 で処理する構成を動的再構成可能なデバイスである PCA-1 に実装した。その結果、従来研究での実装よりも高速な処理が可能であることを確認した。

キーワード　直交変換, 高速アダマール変換, 動的再構成, PCA

Proposal of N-dimensional Fast Hadamard Transform Algorithm and Implementation on Dynamically Reconfigurable Device

Hiroaki TAKAHASHI, Junji KITAMICHI, and Kenichi KURODA

The University of Aizu, Graduate School of Computer Science and Engineering

E-mail:　{m5061219, kitamiti, kuroken}@u-aizu.ac.jp

Abstract　We propose architecture of Fast Hadamard Transform (FHT), one of the orthogonal transform for digital signals. Our proposed architecture can reduce parallelism of FHT butterfly to $1/2^n$ dynamically. Also, we describe a design method for implementing proposed architecture on dynamically reconfigurable devices. Therefore, our FHT system has dynamical alterability of dimension, number of signals, and parallelism. This alterability enables the system to adapt itself to outside requirements changing. We implemented the architecture on a dynamically reconfigurable device PCA-1 with single parallelism. As the result of implementation and evaluation, our system has achieved higher speed performance than our previous work.

Keyword　Orthogonal Transform, Fast Hadamard Transform, Dynamically Reconfigure, PCA

1. まえがき

ディジタル信号の直交変換は、多くの信号処理分野で利用される。信号は直交変換することで周波数成分に変換され、その特徴が得やすくなる。このため、目的とされる処理が元のデータに対して行うよりも効率的になる場合が多い。この応用としては音声・画像の解析や認識、データ圧縮などがある[1][2]。直交変換をデータ量の多い画像に対して行う場合は、演算量を少なくするために画像を小さなブロック(8×8 や 16×16 ピクセル)に分割して処理する場合が多い。この場合、変換値のデータはその小ブロック内の特徴しか表現しておらず、原画像全体の特徴を得ることは困難である。特に画像全体の特徴が重要である医療分野では、原画像の直交変換処理を高速に行うことが必要になる[6]。そのため、ハードウェア化による高速な処理が望まれる。一方、画像認識では、原画像を大局的に領域分割し、分割された画像毎に認識処理を行う場合がある[3]。この場合、認識システムへの入力画像サイズは動的に変化するため、直交変換を行うブロックサイズは柔軟に変更できることが望ましい。

本論文ではこれらの要求を満たす高速かつ柔軟な直交変換システムについて述べる。まず高速な処理を可能とするために、直交変換の中でも演算が加減算のみで行えるアダマール変換[4]に注目した。このアダマール変換には高速化アルゴリズムが存在し、バタフライ演算により加減算の回数を減らすことで高速化を実現できる[5]。本論文ではこのバタフライ演算の並列度を $1/2^n$ に減らして処理可能なアーキテクチャを提案し、高速アダマール変換を少ないハードウェアリソースで実現する。また、提案するアーキテクチャは文献[5]の構成とは異なり、直交変換による変換値を交番数順に出力する。このため、直交変換後に想定される圧縮や認識の処理が容易になると期待できる。次に、提案する高速アダマール変換に柔軟性を持たせるために、これを動的再構成可能デバイス上に実装する。これにより、外部からの動的な要求変化に応じて、処理次元、信号数および並列度を柔軟に変更可能としている。動的再構成可能なデバイスとしては、PCA(Plastic Cell Architecture)-1[7][8]を使用した。PCA-1 を使用して提案アーキテクチャの実装を行った結果、前回の実装[10]と比較して高速な処理が可能であった。また、動的な処理内容変更に伴う再構成について考察を行う。

2. N 次元高速アダマール変換アルゴリズムの提案

本章では、アダマール変換を実現する行列から高速アダマール変換アルゴリズムを導く。次に、高速アダマール変換アルゴリズムから任意の2のベキ乗組のハードウェアリソース上へのマッピングおよび、動的再構成のための構成について述べる。

以下で述べるアダマール変換は、簡単のため入出力の次元が1または2であり、また入出力の各次元の要素数は等しく2のベキ乗の数であるものを対象とする。

[定義1] アダマール変換

入出力のデータの次元数がN、各次元の要素数はすべて同じM（ただしMは2のベキ乗の数）であるようなアダマール変換は、N=2の場合が式1、N=1の場合が式2により定義される。入力を $x(i_1,...,i_N)$,出力を $X(o_1,...,o_N)$と表す。

$X(o_1,o_2)=H2(N)x(i_1,i_2)$ (式 1)

$X(o_1)=H_1(N)\ x(i_1)$ (式 2)

$$H_1(1)=\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$$

$$H_1(2K)=\begin{bmatrix}H_1(k) & H_1(k)\\ H_1(k) & -H_1(k)\end{bmatrix}$$

$H_1(2K)$の要素 $x_{ij}(0\leqq i,j\leqq 2k-1)$の値は、iおよびjを2進表記し、各桁で両方とも1である桁数が奇数のとき-1、それ以外のとき1となる。$H_2(N)$は、2次元入力のある次元方向に $H_1(N)$を施し、得られた2次元データに対してさらに別次元方向に $H_1(N)$を施した行列として表される。 □

アダマール変換の逆変換 $H_M^{-1}(N)$も同様に $(1/M^N)H_M(N)$で表され、M^N による除算を除けば、アダマール変換と同形である。例えば、$H_1(4)$は、

$$\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{bmatrix}$$ と表される。

高速アダマール変換アルゴリズム

このアダマール変換に対する高速アダマール変換アルゴリズムは以下のように再帰的に定義される。

$H_1(2)(i_1,i_2)= [i_1+i_2,\ i_1-i_2]$

$J_1(2)=H_1(2)$

$J_1(2k)(i_1,...,i_k,i_{k+1},...,i_{2k+1})=$
$[A(1)+B(1),A(1)-B(1),...,A(2k-1)+B(2k-1),A(2k-1)-B(2k-1),$
$A(2)+B(2),A(2)-B(2),...,A(2k)+B(2k),A(2k)-B(2k)]$

$A=J_1(k)(i_1,...,i_k),\ B=J_1(k)(i_{k+1},...,i_{2k+1})$

$H_1(2k)(i_1,...,\ i_k,i_{k+1},...,i_{2k+1})=CROSS[J_2(2k)]$

CROSSは、再帰的に構成された $J_1(2k)$のデータ系列順を正しい $o_1,...,o_{2K}$に整列する関数である。 □

例えば $H_1(8)$に対して $CROSS[x_0,x_1,x_2,x_3,x_4,x_5,x_6,x_7]$ $=(x_0,x_4,x_2,x_6,x_1,x_5,x_3,x_7)$である。$J_1(k)$から $J_1(2k)$を求める計算は、$J_1(2)$（入力に対する加減算）の機能を用いればK個で実現可能である。通常CROSSは恒等写像として実現され、出力は交番数順で得られる。

2次元の高速アダマール変換は、アダマール変換の定義により、$J_2(2)$は $J_1(2)$をある次元方向に施し、その結果に対して、別の次元に対して $J_1(2)$による変換を行う。また $J_2(2K)$の再帰的な構成も同様に2回同じ変換を行うことにより定義される。従って、2次元のハードウェアリソースは、1次元の場合の2倍となる。以下でのアルゴリズムでは、1次元の場合について述べ、2次元の場合については省略する。上記の構成は図1のように再帰的に示される。

$J_N(M)$の演算は、基本回路として $J_1(2)$を用いると、$N*M*\log_2 M$ 個で実現できる。

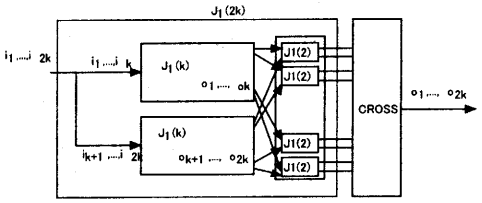

図 1 高速アダマール変換のハードウェア構成

次に、動的再構成のために、高速アダマール変換アルゴリズムを述べる。

動的再構成向き高速アダマール変換アルゴリズム

$H'_1(2)(i_1,i_2) = [i_1+i_2,\ i_1-i_2]$

$H'_1(2k)(i_1,...,i_k,i_{k+1},...,i_{2k+1}) =$
$CROSS2[H'_1(2)(A(1),B(1)),...,H'_1(2)(A(k),B(k))]$
$A = H'_1(k)(i_1,...,i_k),\ B = H'_1(k)(i_{k+1},...,i_{2k+1})$
$CROSS2[x_0,...,x_{2k-1}] =$
$(x_0,x_1,x_3,x_2,...,x_{4n},x_{4n+1},x_{4n+3},x_{4n+2},...)$ □

上記の再帰的定義を用いることにより、$H'_1(2)$およびCROSS2を実現するハードウエアがあれば、出力順が交番数順となる高速アダマール変換が実現できる。この構成は図2になる。

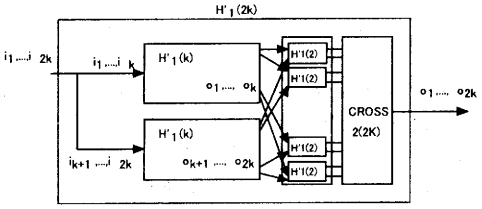

図 2 動的再構成向き高速アダマール変換の構成

しかしこの構成では $H'_1(2)$の数が $N*M*\log 2M$ 個必要である。次にハードウェア量を削減するために、ハ

ードウェア量を 1/2 に、パラレルに入力されるデータをシリアルに入力させることによりハードウエア量を減らすことを考える。アダマール変換 $X(o_1,\ldots,o_{2N})=H'_1(2N)x(i_1,\ldots,i_{2N})$ に対して、入力を $i_1,\ldots,i_{2N}$ と $i_1,\ldots,i_{2N}$ の2列に、出力 $o_1,\ldots,o_N$ と $o_{N+1},\ldots,o_{2N}$ の2列としたとき、入出力ベクトルの次元が N であり、$(H"_1(N)x(i_1,\ldots,i_N),\ H"_1(N)x(i_{N+1},\ldots,i_{2N}))$ が $X(o_1,\ldots,o_{2N})$ と等しくなるような $H"_1(N)$ の構成方法を述べる。

動的再構成向き高速アダマール変換アルゴリズムの 1/2 並列度での実現

$H'_1(2)$および CROSS2 からなる $H'_1(k)$の構成において $H'_1(k/2)$と $H'_1(2)$を考える。$H"_1(k)$は、$H'_1(k/2)$と $H'_1(2)$の間に CROSS3 を挿入した構成である。CROSS3 は 2 つの入力列$[i_1,\ldots,i_{k/2}][i_{k/2+1},\ldots,i_k]$に対して、2 つの出力列$[i_1, i_{k/2+1},i_2, i_{k/2+2},i_3,\ldots][\ \ldots,i_{k/2},i_k]$を出力する。

さらに並列度を下げる場合は、上記の構成を再帰的に実行させる。その結果、CROSS3 は入力系列 $[i_1,\ldots,i_n],[i_{n+1},\ldots,i_{2n}],\ldots,[i_2m_{n-n+1},\ldots,i_2m_n]$ に対して、$[i_1,i_2m_{n/n+1},\ldots],[i_2,i_2m_{n/n+2},\ldots],\ldots,[i_2m_{n/n},i_{(2m_{n/n})*2},\ldots,i_2m_n]$ を出力する。 □

この方法により、K 組必要であったハードウェア量を一組までの任意の 2 のベキ乗分の 1 の組にすることが可能である。この構成を図 3 に示す。また、$H'_1(2)$、CROSS2 および 3 を必要数分複製あるいは不要分を削除し、CROSS3 の制御を変更することにより任意の並列度に対する動的再構成システムを構築することが可能である。また、入力データ数あるいは異なる次元の構成に対しても CROSS3 の制御を変更することにより対応できる。

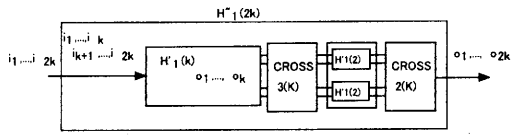

図 3 動的再構成向き高速アダマール変換の 1/2 並列度による構成

上記の構成を再帰的に行って 2 のベキ乗分の 1 にハードウェア構成を減らした場合、CROSS3 は最大その 2 のベキ乗個の入力系列に対して処理を行う必要がある。この変換をハードウェアで構成した場合、2 のベキ乗個の入力データ系列を格納し、指定された順で出力する必要がある。**3.**では、動的再構成デバイス PCA-1 を用いて実装する方法を述べる。

3. 提案アルゴリズムの実装

本章では、**2.**で提案した高速アダマール変換のためのアーキテクチャを動的再構成可能なデバイス上へ実装する方法について述べる。動的再構成可能デバイスとしては PCA-1[8]を利用した。実装では PCA-1 の特徴を活かしているため、まず PCA-1 について説明し、その後回路設計について述べる。

3.1. 動的再構成可能デバイス PCA-1

使用した動的再構成可能デバイス PCA-1 について説明する。PCA-1 を用いることで、外部からの動的な要求変化に応じて高速アダマール変換における処理次元、信号数、または並列度を柔軟に変更することを可能としている。

PCA-1 は、論理回路および記憶要素を実現する可変部と、デバイス外部および可変部間の通信を制御する組込部との二重構造を持つ PCA セルが 2 次元に規則的に並んだ構造になっている(図 4)。

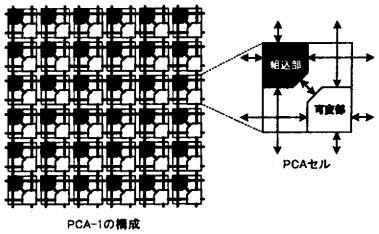

図 4 PCA-1 および PCA セルの構成

可変部 PCA-1 の可変部は 8×8 個の基本セルによって構成されている。基本セルは 4 入力 1 出力の LUT (Look-Up Table)を東西南北に 1 つずつ、計 4 つの LUT で構成されており、隣接する基本セルと接続している。可変部および基本セルの構成を図 5 に示す。可変部に構成される論理回路は、この LUT に設定情報を書き込むことによって実現されている。可変部は隣接する PCA セルの可変部と接続されており、複数の可変部を使った回路を構成することが可能である。また、1 つの可変部は 4 ビットを 1 ワードとした 1024 ワードのメモリとしても使用可能である。

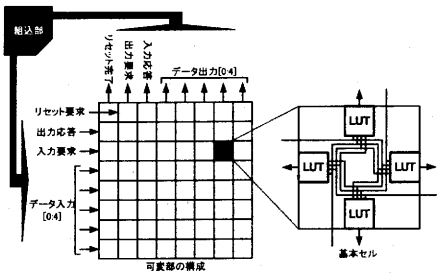

図 5 可変部および基本セルの構成

組込部 組込部も隣接する 4 つの PCA セルの組込部と接続されており、隣接する組込部および自身の PCA セ

ル内の可変部との間で非同期ハンドシェイクによる通信を行う。通信は5ビットを1ワードとするストリームで行われ、組込部への各入力は組込部で処理すべき命令、あるいは可変部へ転送されるべきデータである。PCA-1における組込部の命令は、隣接するPCAセルへの経路設定命令、可変部への転送命令、メモリの書込み・読出し命令、可変部との接続命令など12種類の命令がある。PCA-1はこれらの命令系列を外部から読み込み、自らメモリを書き換えることで動的再構成を行うことができる。組込部は複数の組込部間でのデータ転送の機能を利用して、組込部上の通信経路にデータを保持することが可能である。以下、この機能をバッファリング機能と呼ぶ。

3.2. 高速アダマール変換回路の設計

提案した高速アダマール変換の回路設計について説明する。今回は1次元および2次元の高速アダマール変換を提案アーキテクチャによる並列度1での処理回路を設計、実装した。提案した高速アダマール変換は2点アダマール変換回路とデータの並び替えのための回路等、全4種類の回路で実現可能である。

以降では、まず提案する高速アダマール変換を構成するための4種類の基本回路について説明する。その後、その回路を用いた1次元および2次元高速アダマール変換の構成についてそれぞれ説明する。また、本論文で設計する高速アダマール変換回路は、動的再構成可能デバイスを利用することにより、その時々の状況や必要に応じて処理次元、信号数、並列度を動的に変更できる。これについては**5.**で述べる。

3.2.1. 高速アダマール変換の構成要素

提案した高速アダマール変換の構成要素となる回路は、処理する次元数および並列度に関わらず同じ回路を用いることができる。本章では、その構成要素個々の回路について説明する。

構成要素となる回路は、2点アダマール変換を行う回路(2pWHT、**2.**で述べた $H'_1(2)$)、その出力をそのままあるいは交差させて出力する回路(交差回路、**2.**で述べたCROSS2)、1入力を2出力に分配する回路(分配回路)、および2入力を1出力に統合する回路(統合回路)の4種類の回路である。統合および分配回路は**2.**で述べたCROSS3である。ただしそれぞれの回路は、その回路内の制御方法が異なる回路が必要である。以下では、使用目的と各回路の詳細な動作について説明する。

2pWHTは、2つの入力A、Bに対して2点アダマール変換、すなわち$A+B$と$A-B$の演算を下位ビットから行い、その計算結果をそれぞれ下位ビットから出力する回路である。2pWHTは処理ビット数に応じて加減算の繰り返しの為の制御部分が異なる。例えばA、Bがそれぞれ8ビットで表現される場合、PCA-1ではデータが4ビットずつのストリームとして入力されるので下位および上位の2回の加減算を行う制御となる。回路内部では4ビットを並列に加減算する。

交差回路は、2pWHTからのデータをそのまま、あるいは交差させて出力する回路である。この回路は提案する高速アダマール変換による変換値を交番数順で出力するために使用する。

分配回路は、1つの入力と2つの出力を持ち、一定数の入力毎に出力ポートを切り替えて入力データを分配出力する。統合回路は逆に、2つの入力と1つの出力を持ち、一定数の入力毎に入力ポートを切り替え、そのデータを出力する。これら2つの回路を組み合わせることで、パラレルデータとシリアルデータを変換することができる。この組み合わせによりストリームデータの順番を入れ替え、直後の2pWHTに正しい順番で処理させる。

以上の4種類の回路を組み合わせて提案アーキテクチャによる高速アダマール変換が実現できる。これらの個々の回路は、スループットが同程度になるように設計した。そのため、これらの組み合わせによる構成ではパイプラインが途中で詰まらない処理となる。

なお、今回設計する高速アダマール変換回路における変換値データは、すべて変換後のデータが取りうる範囲を表現可能なビット数で出力する。例えば、1ピクセルが4ビット(0から15の値)で表現された8×8ブロックサイズの画像データに対して2次元アダマール変換を行う場合を考える。このときの最大値は15×8×8=960、最小値は-480であり、この範囲を表現可能なビット数で変換値を出力する。そのため、入力ビット数は出力ビット数と同じビット数に合わせ、上位ビットに0を付加する。

PCA-1ではデータを4ビットストリームで扱うが、本設計でも、すべての回路への入力データは4ビットのシリアルデータとして入力される。例えば2pWHTにおいて、12ビット演算を行う場合、A、Bへの入力はそれぞれ4ビットのデータが3回入力される。

3.2.2. 1次元高速アダマール変換回路の設計

3.2.1.で述べた4種類の回路を用いて1次元高速アダマール変換を構成する。例として8点入力の1次元高速アダマール変換回路を図6に示す。

2pWHT以外の回路ブロック内の数字は制御の数を示しており、2ならばデータ2つ分の制御を行っている。ただし、1つ分のデータは変換値を表現可能なビット数に依存しており、8点アダマール変換の場合は1データが8ビットとなる。これが16点アダマール変換の場合、変換値を表現可能なビット数にする影響で、1データは12ビットになる。そのため、同じ分配2回路

でも、変換値のビット数に応じて制御が異なる回路に変更する必要がある。

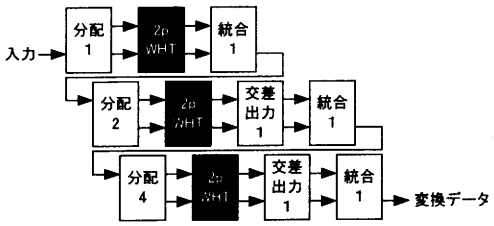

図 6　1 次元高速アダマール変換(8 点)

ストリームデータの並び替え時には一時的なバッファリングが必要になる。例えば分配 4 では、上の経路に 4 つ分の入力データを流し、続く 4 つ分の入力データを下の経路に流すという動作を繰り返す。分配 4 回路の直後の処理である 2pWHT は実行に必要な上下のデータがそろわない限り動作しない。そのため、上の経路に分配された 4 つ分のデータは、2pWHT の下の入力データが到達するまで待つことになり、2pWHT が動作するまで 4 つのデータを一時的に保持する必要がある。本設計では、PCA-1 組込部のバッファリング機能を用いてデータの一時的な保持を行っている。このため、分配 4 と 2pWHT の間はこの 4 つのデータが保持可能な長さの経路(組込部)が必要となる。

3.2.3. 2 次元高速アダマール変換回路の設計

2 次元高速アダマール変換も規則的に構成できる。例として 8×8 点のデータを処理する 2 次元高速アダマール変換回路の構成を図 7 に示す。2 次元高速アダマール変換は、入力データの x 軸方向、y 軸方向についてそれぞれ 1 次元高速アダマール変換を行っているので、$n \times n$ 点 2 次元高速アダマール変換の構成では、n 点の 1 次元高速アダマール変換と比較して 2pWHT の数が 2 倍になる。

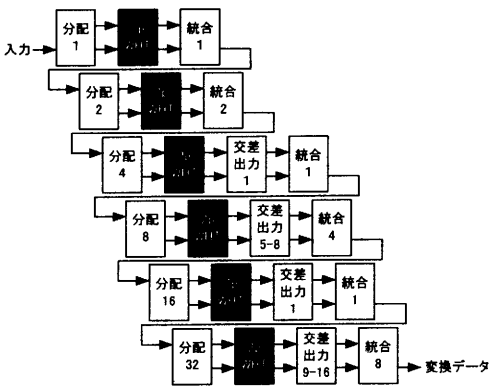

図 7　2 次元高速アダマール変換の構成(8×8 点)

4. 評価

今回設計した高速アダマール変換回路を評価するために、PCA-1 シミュレータである PCASIM II[9]を使用した。本章で示す回路構成時間および処理時間はこのシミュレータによるものである。また、シミュレーションの際、構成情報、入力信号データ等のすべての PCA-1 への入力は 1 つの外部ホストから 1 つのポートのみを通り 4 ビットのストリームデータとして入力されるものとする。また、入力信号データのビット数は 1 次元および 2 次元高速アダマール変換、いずれの場合も 4 ビットとした。

今回設計した回路は、1 次元高速アダマール変換では 8、16 点が処理可能な回路、2 次元高速アダマール変換では 4×4、8×8 点が処理可能なものである。これらの回路の構成に使用する可変部および組込部のセル数はそれぞれ表 1 の通りである。

表 1　各次元の高速アダマール変換回路のセル数

	1 次元		2 次元	
	8 点	16 点	4×4	8×8
可変部	38 セル	52 セル	48 セル	76 セル
組込部	102 セル	161 セル	189 セル	564 セル

今回組込部に構成する経路は十分な最適化を行っていないため、これ以下のセル数にすることが可能である。今回の設計における使用セル数の中では、可変部に構成される回路のセル数に比べて通信経路に使用するセル数(組込部)が多い。これは **3.2.2.**で説明した、データの並び替え時のデータ保持を組込部のバッファリング機能を利用しているためである。処理信号数、またはブロックサイズが増加するに従ってこのバッファリングのための経路は長くなる。この組込部のセル数の増大は、データの並び替え時のデータ保持を組込部のバッファリング機能から可変部のメモリに変更することで解決できる。ただし、メモリ操作による制御の複雑化によってスループットの低下が予想される。

これらの回路を PCA-1 上に構成する時間はそれぞれ表 2 の通りである。

表 2　各次元の高速アダマール変換回路の構成時間

	1 次元		2 次元	
	8 点	16 点	4×4	8×8
構成時間(sec)	641 μ	859 μ	800 μ	1280 μ

この時間は、最初の構成データが PCA-1 上にダウンロードされ始めてから、すべての構成が終了するまでの時間である。

また、処理時間に関する結果は表 3、4 の通りである。

表 3　1次元高速アダマール変換の処理時間

	信号数	
	8 点	16 点
レイテンシ(sec)	1.71 μ	4.42 μ
スループット(nibble/sec)	13.87 M	14.15 M
全信号分の処理時間(sec)	1.15 μ	3.39 μ

表 4　2次元高速アダマール変換の処理時間

	ブロックサイズ	
	4×4	8×8
レイテンシ(sec)	4.63 μ	16.58 μ
スループット(nibble/sec)	14.13 M	14.08 M
全信号分の処理時間(sec)	3.40 μ	13.64 μ

レイテンシは、最初に処理するデータが PCA-1 に流れ始めてから、処理後の最初の変換値が PCA-1 から出力されるまでの時間とした。スループットはパイプラインの流れが安定した状態での単位時間当たりの出力データ数である。単位の nibble は、PCA-1 では 4 ビットデータ 1 つ分である。全信号の処理時間は、パイプラインの流れが安定した状態において、全信号分(ブロックサイズが 8×8 の場合は、このブロック内のすべてのデータ)の処理にかかる時間である。従来研究のアダマール変換回路[10]は、今回のパイプライン設計ではなく、同じ加減算器による演算の繰り返しで変換処理を行う構成であった。処理時間は、従来研究の結果に比べて極めて短時間である。例えば 8×8 サイズのデータを処理する時間では、従来構成の場合よりも約 440 分の 1 の時間で処理可能であることがわかった。これは、1 ピクセル 4 ビットで 1280×1024 サイズの画像データを約 280msec で直交変換可能な時間である。

5. 動的な処理内容変更における考察

動的再構成可能デバイスを用いることにより、提案する高速アダマール変換における処理信号数を動的に変更することが可能である。例えば図 6 の 8 点入力 1 次元高速アダマール変換の構成の最後の統合 1 回路の後に、分割 8、2pWHT、交差出力 1、および統合 1 の回路を動的に付け足すことで 16 点アダマール変換処理を実行可能な構成に変更できる。一般に 2^n 点の入力を処理する 1 次元高速アダマール変換は、n 組の分配、2pWHT、交差、および統合回路で実現できる。ただし入力信号数が異なると、変換値のビット数が異なるため、各構成回路の制御部分を変更する必要がある。

図 7 の 8×8 点を処理する 2 次元高速アダマール変換の構成を 16×16 点を処理可能な構成に再構成する場合では、図の統合 8 の出力の後に、分割 64、2pWHT、交差出力 1、統合 1、その後、分割 128、2pWHT、交差出力 17-32、統合 8 の回路をつづけて構成し、さらに制御の異なる回路を上書きすれば変更が可能である。$n \times n$ 点の信号を処理する 2 次元高速アダマール変換は、$2n$ 組の分配、2pWHT、交差、および統合回路で実現できる。

さらに、次元数が異なる入力データに対しては、その次元の処理が可能な構成へと動的に再構成することで変更可能であり、変換回路の並列度の変更をすることも可能である。

6. むすび

本論文では、ディジタル信号の直交変換処理の一つであるアダマール変換を、処理次元、信号数および並列度を動的に変更可能な N 次元高速アダマール変換のアーキテクチャとして提案した。また、提案アーキテクチャによる並列度 1 の場合の高速アダマール変換回路を、動的再構成可能デバイス PCA-1 上に実装した。実装の結果、処理時間においては、8×8 サイズのデータを 2 次元アダマール変換する場合を我々の従来研究による構成での処理時間と比較した場合、約 440 倍もの高速化を実現できた。また、動的な処理内容の変更における処理回路の再構成についての考察を行った。

今後の課題としては、組込部でのバッファリングのための経路を短くし、実装面積を減らすことが挙げられる。また、動的な構成変更も実装し、提案手法の評価を行う予定である。

文　献

[1] 安居院 猛, 長尾 智晴, 画像の処理と認識, 昭晃堂, 1992.

[2] 田村 秀行, コンピュータ画像処理入門, 総研出版, 1985.

[3] 森江 隆, 岩田 穆, "脳機能に学ぶ画像認識集積システム," 信学技報, VLD2002-51, pp.67-78, Jun. 2002.

[4] K. Katayama, A. Iwata, T. Morie, and M. Nagata, "An Hadamard Transform Chip Using the PWM Circuit Technique and Its Application to Image Processing," IEICE TRANS. ELECTRON., Vol.E85-C, No.8, pp.1596-1603, Aug. 2002.

[5] K. Komatsu and Sezaki, "Lossless 2D Discrete Walsh-Hadamard Transform," IEEE ICASSP2001, vol.3, May. 2001.

[6] V. G. Ruiz and I. G. García, "On the Computation of the Fast Hadamard Transform for Large Amounts of Data," VII National Symposium on Pattern Recognition and Image Analysis. 1997.

[7] 名古屋 彰, 小栗 清, "プラスチックセルアーキテクチャ(PCA)技術の概要," NTT R&D, Vol.49, No.9, pp.513-517, Sept. 2000.

[8] 中田 広, 伊藤 秀之, 小西 隆介, "完全非同期回路による PCAハードウェアの設計・評価," NTT R&D, Vol.49, No.9, pp.518-526, Sept. 2000.

[9] 小西 隆介, 伊藤 秀之, 中田 広, 小栗 清, 名古屋 彰, "PCA の実現を支援するソフトウェア," 第 16 回パルテノン研究会資料集, pp.3-12, May. 2000.

[10] 高橋 宏章, 北道 淳司, 西村 憲, 黒田 研一, "ブロックサイズおよび並列度を動的変更可能なアダマール変換回路の設計," 第 1 回リコンフィギャラブルシステム研究会論文集, pp.157-164, Sept. 2003.