

動的再構成 ALU を搭載した画像認識プロセッサ — 高速フーリエ変換・逆変換に適した再構成コンピューティング —

宮本 直人[†] 小谷 光司[†] 丸尾 和幸[‡] 大見 忠弘[§]

[†] 東北大学大学院工学研究科 〒980-8579 宮城県仙台市青葉区荒巻青葉 6-6-10

[‡] アドバンテスト研究所 〒989-3124 宮城県仙台市青葉区上愛子字松原 48-2

[§] 東北大学未来科学技術共同研究センター 〒980-8579 宮城県仙台市青葉区荒巻青葉 6-6-10

E-mail: [†] {miyamoto, kotani, ohmi}@fff.niche.tohoku.ac.jp, [‡] maruo@atl.advantest.co.jp

あらまし 位相限定相関法(POC)アルゴリズムを実装した画像認識プロセッサを設計した。本プロセッサには動的に再構成可能な算術演算器(ALU)が搭載されている。POC アルゴリズムを再構成コンピューティングに適した形に並び替えることにより、本プロセッサは二次元 512x512 ピクセルの画像認識を 105.2msec, 310.9mW で実行することができる。この消費電力の値は、現在までに提案されている他の POC プロセッサの 11.3 分の 1 である。

キーワード 動的再構成, 算術演算器, 位相限定相関, 高速フーリエ変換

An Image Recognition Processor Using Dynamically Reconfigurable ALU — Reconfigurable Computing Suitable for Algorithms Using FFT and IFFT —

Naoto MIYAMOTO[†] Koji KOTANI[†] Kazuyuki MARUO[‡] and Tadahiro OHMI[§]

[†] Faculty of Electronic Engineering, Tohoku University 6-6-10 Aoba, Aramaki, Aoba-ku, Sendai, 980-8579 Japan

[‡] Advantest Ltd., 48-2 Aza-Matsubara, Kamiyashi, Aoba-ku, Sendai, 989-3124 Japan

E-mail: [†] {miyamoto, kotani, ohmi}@fff.niche.tohoku.ac.jp, [‡] maruo@atl.advantest.co.jp

Abstract An image recognition processor utilizing phase only correlation (POC) algorithm is proposed. Arithmetic logical unit (ALU) in this processor can be re-configured dynamically. By arranging POC algorithm to the form suitable for reconfigurable computing, the proposed processor can perform two-dimensional 512x512 pixels image recognition within 105.2msec and 310.9mW. This power consumption is 11.3 times lower than that of a previously reported work with same execution time.

Keyword Dynamically Reconfigurable, ALU, Phase Only Correlation, Fast Fourier Transform

1. はじめに

位相情報に着目した画像認識アルゴリズムが近年脚光を浴びている。その代表である位相限定相関法(POC)アルゴリズムは、他の認識アルゴリズムと比べて高い認識能力、ロバスト性および 1/100 サブピクセルの解像度を持つと言われている[1,2]。

POC は位相情報を扱うため、高速フーリエ変換(FFT)および高速逆フーリエ変換(IFFT)が必要である。しかし、どちらの変換も非常に長い実行時間と高い消費電力が要求される[3]。また、FFT や IFFT の他にも、POC プロセッサには位相算出や相関を計算することが求められる。もし、これらの計算を独立した個別の算術演算器(ALU)で実行すると、プロセッサ面積が大きくなるだけでなく、計算ごとに演算に必要なデータを適宜 ALU に移さねばならないため長い実行時間と多くのメモリアクセス回数が必要となる。

POC のデータフローは分岐のない一本道であるため、アルゴリズム中で出現する演算はひとつずつしか

実行されない。しかし、FFT や IFFT はメインメモリにアクセスする回数が他の演算よりも多く、多数の大容量メモリをプロセッサ上に実装しない限りパイプライン化は難しい。しかも、FFT の実行時間は位相算出や相関計算に比べて長く、位相算出や相関計算の開始は FFT の実行終了を待つ必要がある。

動的再構成技術に対する関心が、近年、非常に高くなってきている[4]。処理内容に合わせて回路を瞬時に最適な構成に変化させることが可能な動的再構成プロセッサは、低面積化・低消費電力化が期待できる。

本論文では POC を実装した動的再構成プロセッサについて述べる。第 2 章では POC アルゴリズムの概要に触れ、第 3, 4 章では同時に実行されない複数の演算器を資源共有化した動的再構成 ALU、第 5 章では動的再構成コンピューティングに適した POC アルゴリズムへの変形について述べ、第 6 章では設計したプロセッサを評価し、最後に結論を述べる。

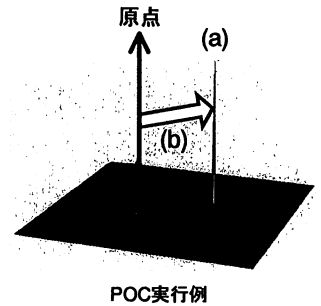
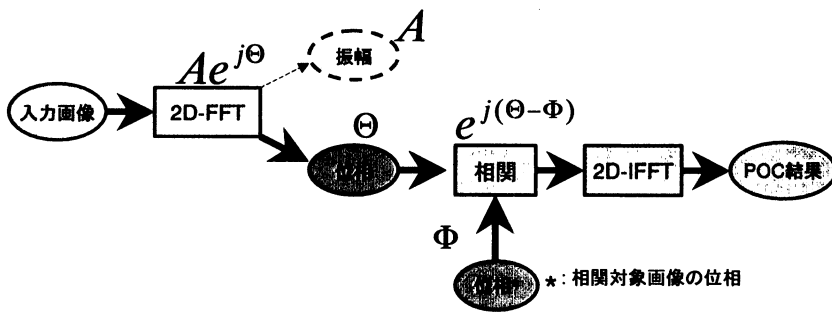


図1 位相限定相関法(POC)アルゴリズムと実行例

2. 位相限定相関

POCアルゴリズムは、入力画像と相関対象画像の相関を取る画像認識アルゴリズムの1種である。フーリエ変換から得られる画像の位相情報は振幅情報よりも画像認識においては重要な役割を演じる。この理由は、画像中に存在する物体の形状や位置の情報は、フーリエ変換の結果、位相情報に集約されるためである[5]。逆に、物体の明るさや回転などの情報は振幅情報として反映されるが、この情報は時として誤認識の原因にもなる。

POCでは、振幅情報を一定の値に置換することで削除し、位相情報のみを用いて画像の相関を取る。図1にPOCのアルゴリズムとその実行例を示す。POCの際立った特長は2つある。1つは、(a)入力画像と相関対象画像の中に同一オブジェクトがある場合、クロネッカーの Δ 関数が出現すること。もう1つは、(b)クロネッカーの Δ 関数のピークの二次元座標が、入力画像と相関対象画像の中のオブジェクトの位置のずれに正確に反映されることである[1,2]。

3. 動的再構成ALU

図1に示すように、POCアルゴリズムは以下の4つのステップから構成されている。

1. 二次元フーリエ変換(2D-FFT)
2. 振幅情報と位相情報の分離(振幅情報の削除)
3. 相関(位相の合成)
4. 二次元逆フーリエ変換(2D-IFFT)

FFTおよびIFFTは、時間あるいは空間領域のデータを周波数領域のデータに変換したり、逆に周波数領域のデータを時間あるいは空間領域のデータに変換する。FFTとIFFTには共にバタフライ演算が必要である。振幅情報を算出するためには、積和演算(Multiply

Accumulation: MAC)および平方根が必要である。位相は、FFTの結果を振幅で割った商である。相関すなわち位相の合成は、入力画像の位相 θ を相関対象画像の位相の複素共役 $-\phi$ だけ回転させることにより得られる。

C言語スタイルで記述したPOCアルゴリズムを図2に示す。アルゴリズム中に4つのfor文がある。これらは、図1に示すように、分岐のない直線的な前後関係があり、直前のfor文が終了しない限り次のfor文が開始できない。したがって、機能固定型ALUを複数用いてPOCを演算処理する場合、ALUはどの時刻を見ても1つしかアクティブになっていない。

一方、動的再構成コンピューティングでは、そのようなデータハザード状態でストールしているALUは存在しない。なぜならば、単一の動的再構成ALUはアルゴリズムに必要なすべての演算器構成に数ナノ秒で再構成することができ、データの状態に合わせて実行すべき演算機能を次々に変化させていくことができるからである。

```

N = number of vertical pixels of an image;
M = number of horizontal pixels of an image;

for (int i=0; i<N; i++)
  for (int j=0; j<M; j++)
    N-point 2D-FFT(); // (i) バタフライ演算

for (int k=0; k<M; k++)
  Phase Calculation(); // (ii) MAC, (iii) 平方根,
                        // (iv) 除算

for (int l=0; l<M; l++)
  Correlation(); // (v) 複素共役,
                 // (vi) 複素乗算

for (i=0; i<N; i++)
  for (j=0; j<M; j++)
    N-point 2D-IFFT(); // (vii) バタフライ演算

```

図2 C言語スタイルで記述したPOCアルゴリズム

動的再構成 ALU の設計で重要なことは、『同時に実行されない複数の演算器間でハードウェア資源を共有化すること』である。

図 2 に示すとおり、POC には(i)から(vii)までの七種類の同時に実行されない演算器がある。この中で回路規模が最大の演算器は図 3(a)に示すバタフライ演算器である。バタフライ演算器を土台として、図 3(b)に示すような動的再構成 ALU を設計した。この動的再構成 ALU はバタフライ演算器と比較して、幾つかのマルチプレクサと信号の値を論理 0 に固定するための論理積回路が追加されており、また、2 個の乗算器が第 4 章で述べる機能融合モジュールに、1 個の減算器が加減算器に置換されている。この BEU ベースの動的再構成 ALU は、図 2 に示した七種類の演算の中から一度にどれか 1 つに再構成することが可能である。

4. 除算器・開平器・乗算器融合モジュール

図 2 に示すように、POC には(i)から(vii)までの七種類の演算器が必要である。(i),(ii),(vi),(vii)は乗算器を含み、開平器と除算器はそれぞれ(iii)と(iv)が必要である。これらの 3 つの演算器は加算器や減算器よりも多くのハードウェアコストがかかる。そこで我々は乗算器、除算器および開平器を資源共有化した動的再構成モジュール (Divider / Square-root / Multiplier

Merged Module : DSM)を開発した。

DSM は粗粒度の動的再構成可能な論理モジュールの一種である。アレイ乗算器、引放し除算器および開平器はいずれも全加算器のアレイから構成されるため、DSM は全加算器アレイをゲートレベルで共有するように設計した。

図 4(a)–(d)は、DSM を構成する 4 種類の基本演算回路を示している。これらの基本演算回路はいずれも 1 個の全加算器、論理積回路および排他的論理和を持つ。この中で論理積回路は乗算に使われ、排他的論理和回路は除算と平方根に使用される。全加算器の入力ポートに幾つかのマルチプレクサを挿入することでハードウェア資源の共有化を図っている。これらの 4 種類の回路は、図 4(e)に示すように、規則性を持つアレイ状に配置される。各マルチプレクサの選択信号 S1 と S2 は回路構成情報に相当し、これらの信号の値に応じて DSM は

- (a) 除算器 if S1=1 かつ S2=0
- (b) 開平器 if S1=1 かつ S2=1
- (c) 乗算器 if S1=0 かつ S2=0
- (d) NOP if S1=0 かつ S2=1

に構成される。

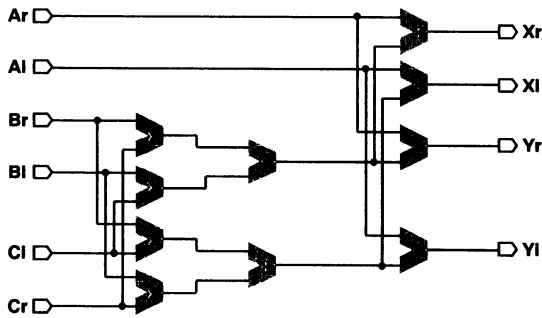


図 3(a) バタフライ演算器(BEU)

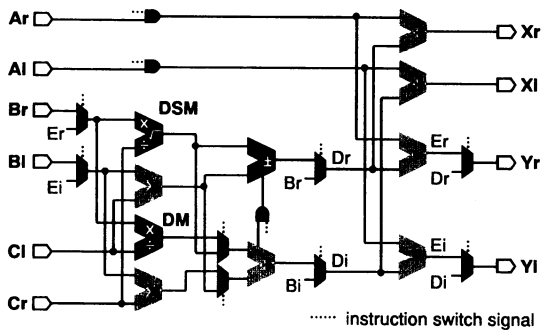
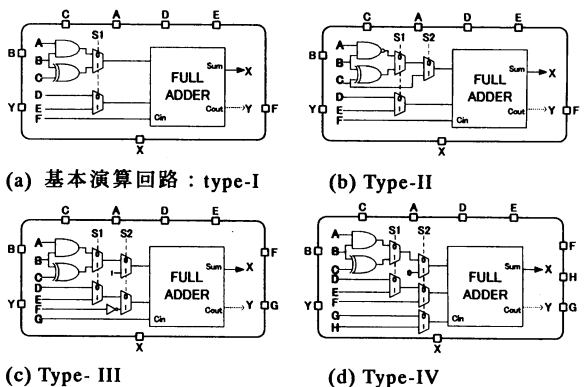
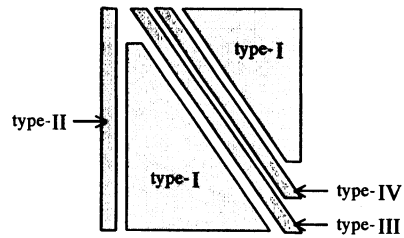


図 3(b) BEU ベースの動的再構成 ALU



(c) Type- III

(d) Type-IV



(e) 基本演算回路のアレイ配置

図 4 乗算器・開平器・乗算器融合モジュール(DSM)の基本構造

ここで、回路構成情報 S1 および S2 は、オンチップのデコーダから生成される。このデコーダは布線論理から成り、約 100 ゲート(チップ全体の 0.05%程度)の規模で、最高 80MHz 以上で動作する。

図 5 に、データバス幅 8 ビットの DSM の例を示す。図中、a[7:0]と b[7:0]は入力データで、m[15:0], q[7:0] および r[7:0]はそれぞれ乗算、除算および平方根の結果である。

資源共有しない単体の乗算器、除算器および開平方器に対する DSM の面積比較を行った。図 6 はその結果を示している。全加算器アレイの資源共有化により、除算器単体(D)、開平方器単体(S)および乗算器単体(S)の面積の合計と比較して、DSM はゲート数では 35%、実際の $0.35\mu\text{m} \cdot 2\text{P3M}$ プロセスで配置配線した面積で

は 24%の削減効果があった。同様にして、演算遅延の測定を行った。資源共有化によるマルチプレクサの追加の影響で、演算遅延は乗算器で 28.8%、開平方器で 26.7%、除算器で 4.2%演算遅延が増加した。

5. POC アルゴリズムの変形

提案する動的再構成 ALU は、POC アルゴリズムに必要なとされるすべての演算の中からいずれか 1 つに数ナノ秒以内で再構成する。そこで我々は、小林氏らが提案した元来の POC アルゴリズム[1,2]を動的再構成 ALU による処理に適した形態に変換した。

図 2 に示す POC アルゴリズムの中で、2D-FFT および 2D-IFFT は、図 7(a)に示すように、それぞれ一次元の FFT(1D-FFT)および IFFT(1D-IFFT)に分解できる。図 7(a)のアルゴリズムを機能固定 ALU で実行する場合には、図 7(b)に示すようなブロック図となるのが一般的である。

それに対して、動的再構成 ALU では、一塊のデータに対してなるべく多くの逐次処理が連続する方が処理の効率が良い。その観点から、図 7(a)のアルゴリズムを図 8(a)のように変形した。二次元の配列データを対象とする 2D-FFT に対して、1D-FFT は一次元のベクトルデータを扱う。2D-FFT から 1D-FFT へ分解することにより、配列処理は、それぞれ独立した行方向および列方向の処理に変わる。2D-IFFT も同様である。

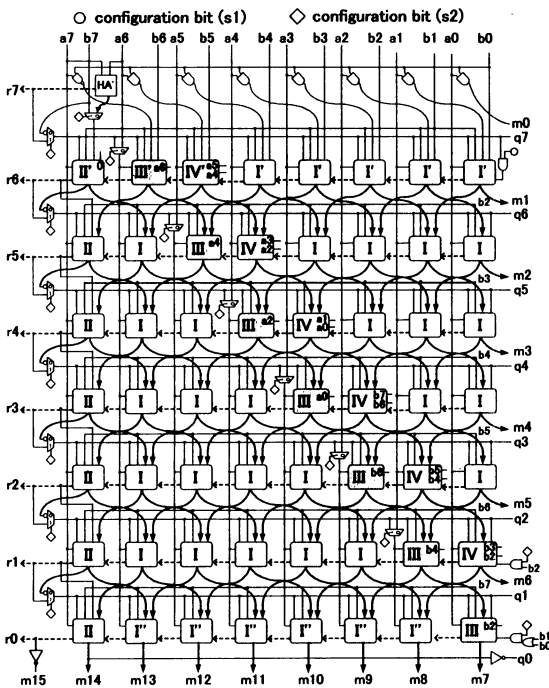


図 5 8-bit DSM の回路図

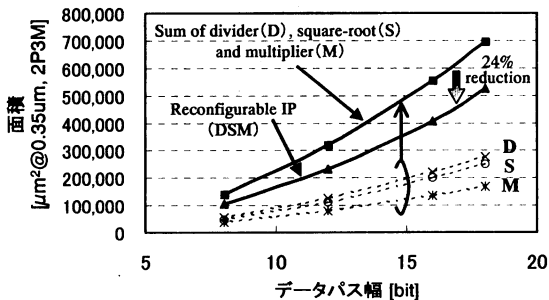


図 6 DSM と個別の演算器の面積比較

```
// 2D-FFT
for (i=0; i<N; i++)
  N-point 1D-FFT(); // 列方向
for (j=0; j<M; j++)
  (A) N-point 1D-FFT(); // 行方向

for (k=0; k<M; k++)
  (B) Phase Calculation();
for (l=0; l<M; l++)
  (C) Correlation();
```

```
// 2D-IFFT
for (j=0; j<M; j++)
  (D) N-point 1D-IFFT(); // 行方向
for (i=0; i<N; i++)
  N-point 1D-IFFT(); // 列方向
```

図 7(a) 2D-FFT, 2D-IFFT を一次元に分解した POC

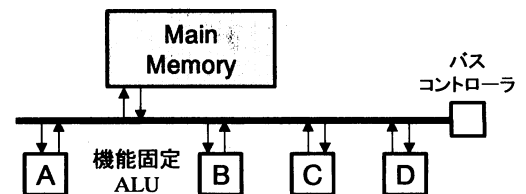


図 7(b) 7(a)を実装した場合のブロック図

```

for (i=0; i<N; i++)
  N-point 1D-FFT();           // 列方向

for (j=0; j<M; j++) {
  // 行方向のすべての処理をまとめる
  (A) N-point 1D-FFT();
  (B) Phase Calculation();
  (C) Correlation();
  (D) N-point 1D-IFFT();
}

```

```

for (i=0; i<N; i++)
  N-point 1D-IFFT();         // 列方向

```

図 8(a) 動的再構成 ALU に適した POC アルゴリズム

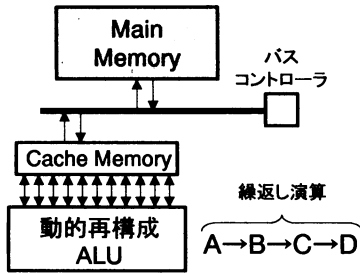


図 8(b) 8(a)を実装した場合のブロック図

一方、位相算出と相関計算は画素単体に対するスカラ演算であり方向には無関係であるが、わざと行方向の 1D-FFT と同じ並び方にする事で行方向のベクトル演算に変形することが可能である。このようにして、すべての行方向のデータに対して 1つの for 文で一括処理できるようにアルゴリズムを変形したものが図 8(a)である。

図 8(b)は、提案する画像認識プロセッサで採用したブロック図を示している。メインメモリとキャッシュメモリおよび 4 個の動的再構成 ALU から構成されている。メインメモリから読み出された 8ワード分のデータは、一旦キャッシュメモリに書き込まれる。その後、キャッシュメモリと動的再構成 ALU の間で繰返し処理が行われ、最終演算結果だけがメインメモリに書き戻される。こうすることで、信号の総伝播距離とメインメモリのアクセス回数を最小化することができるため、低消費電力化が実現できる。

ここで注意が必要なのが、FFT と IFFT のデータオーダである。FFT および IFFT に必要なバタフライ演算器には、時間間引き型(Decimation-In-Time : DIT)と周波数間引き型(Decimation-In-Frequency : DIF)の二種類が存在する。DIT は Reverse Order で入力し、Natural Order で出力される。逆に、DIF は入力が Natural Order で、出力は Reverse Order である。したがって、図 9(a)のように DIT を用いて FFT を行い、

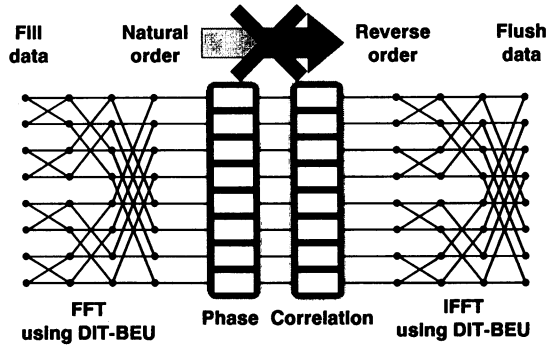


図 9(a) DIT と DIT の組合せ(データオーダが不整合)

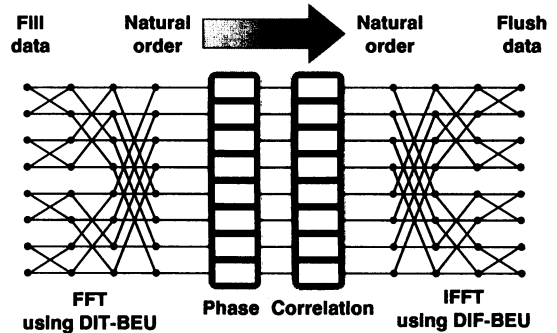
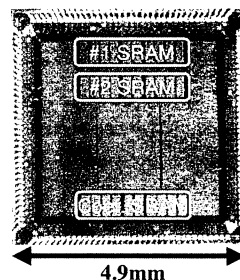


図 9(b) DIT と DIF の組合せ(データオーダが整合)

位相算出と相関計算を行った直後に、再度 DIT を用いて IFFT すると、ビット順序を Natural から Reverse に変換しなければならないためデータの整合性が取れない。整合性を取るためには図 9(b)のように DIT と DIF を組合せる必要がある。こうすれば、データの整合性を取るために一旦メインメモリにアクセスする必要がなくなり、キャッシュメモリと動的再構成 ALU の間で繰返し処理を連続して実行することができる。

6. チップ設計および測定評価

4 個の動的再構成 ALU を搭載した POC プロセッサを設計した。そのチップ写真を図 10 に示す。本プロセッサで採用したメモリバスアーキテクチャは、著者



テクノロジー
0.35μm, CMOS 2P3M
コア面積
3.7mm×3.7mm
解像度
512×512 ピクセル

図 10 動的再構成 ALU を搭載した POC プロセッサ

表 1 提案する動的再構成プロセッサの主な特徴

	Proposed	Cinderella II[1]
プロセス テクノロジー	0.35 μ m	0.35 μ m
供給電圧	3.3V	3.3V
トランジスタ 数	Logic : 400k SRAM : 360k Total : 760k	2,000k
データバス幅	36 bit complex	32 bit
チップ面積	4.9x4.9 mm ²	Not Reported
動作周波数	80MHz	100MHz
解像度	512x512	256x256
消費電力	310.9mW	3.5W
512x512 POC 実行時間	105.2msec	(estimated) 122.2msec

らが提案した二段キャッシュメモリアーキテクチャ [6]であり、FFT や IFFT を用いた低面積・低消費電力二次元画像処理に最適化されている。

表 1 に、我々が提案する動的再構成プロセッサの緒元および現在までに報告されている他の POC プロセッサ Cinderella II[1]との性能比較を示す。提案するプロセッサは、二次元 512×512 ピクセルの POC を 80MHz 動作時に 105.2 ミリ秒、310.9 ミリワットで実行することができる。この消費電力の値は、Cinderella II[1]の 11.3 分の 1 である。低消費電力化した理由は、元来の POC アルゴリズム [1,2]を動的再構成 ALU に適するように変形したことにより、信号の総伝播距離とメインメモリのアクセス回数を最小化することができたためだと考察する。

7. 結論

我々は、POC アルゴリズムを実装した画像認識プロセッサを開発した。本プロセッサには動的に再構成可能な ALU が 4 個搭載されている。この ALU の中には乗算器、除算器および開平方の共通部分である全加算器アレイを資源共有した DSM がある。このように、同時に実行されない複数の演算器間でハードウェア資源を共有化することにより、ゲート数で 35%、面積 (0.35 μ m 2P3M)で 24%の面積削減効果が得られた。

元来の POC アルゴリズム [1,2]を動的再構成コンピューティングに適するように変形した。具体的には、一塊のデータに対して連続的に逐次処理がなされるようにした。この工夫により、本プロセッサは二次元 512 × 512 ピクセルの画像認識を 80MHz 動作で 105.2msec、310.9mW で実行することができる。この消費電力は現在までに提案されている他の POC プロセッサ Cinderella II[1]と比較して、ほぼ同じ実行時間にもかかわらず 11.3 分の 1 の低消費電力を達成することができた。

Acknowledgement

本チップ試作は東京大学大規模集積システム設計教育研究センター (VDEC) を通し、シノプシス株式会社、日本ケイデンス株式会社、ローム(株)および凸版印刷(株)の協力で行われたものである。

文 献

- [1] M. Morikawa, A. Katsumata and K. Kobayashi, "An image processor implementing algorithms using characteristics of phase spectrum of two-dimensional Fourier transformation," The 1999 IEEE International Symp. on Industrial Electronics (ISIE'99), pp. 1208-1213, Slovenia, July 1999.
- [2] 小林孝次, 中島寛, 青木孝文, 川又政征, 樋口龍雄, "位相限定相関法の原理とその応用," テレビジョン学会技術報告, ITE Technical Report Vol.20, No.41, MIP'96-53, NIM'96-75, pp.1-6, July, 1996
- [3] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," Mathematics of Computation, Vol. 19, Issue 90, pp. 297-301, 1965.
- [4] 村上和彰, "世の中はコンフィギュラブル, リコンフィギュラブルへ!", 平成 15 年度 VDEC リフレッシュ教育 VLSI 設計教育コース 3 最先端 VLSI 設計事例 (デジタル), 東京, 2004.
- [5] T. S. Huang, J. W. Burnett and A. G. Deczky, "The importance of phase in image processing filters," IEEE Trans. on Acoustic, Speech and Signal Processing, Vol. ASSP-23, No. 6, 529-542, 1975.
- [6] N. Miyamoto, L. Karnan, K. Maruo, K. Kotani and T. Ohmi, "A 100MHz 7.84mm² 31.7msec 439mW 512-point 2-dimensional FFT single-chip processor," IEICE Trans. on Elec., Vol. E87-C, No.4, pp.502-509, 2004.