

1. はじめに

マルチメディアやインターネットの普及に伴い、実時間での画像処理は非常に重要な技術となっている。画像処理の中でも、画像フィルタ演算はノイズ除去・画像先鋭化・特徴抽出に応用され、様々な画像処理アルゴリズムを実現するにあたり極めて重要な役割を持つ。従って、高度な画像処理アプリケーションには、高速なフィルタ演算が必要不可欠である。

画像フィルタ演算では、あるサイズの係数行列を「カーネル」として定義し、処理対象となる画像から切り出された部分画像とコンボリューション演算を行う。カーネルを画像全体に対して移動させながら、コンボリューション演算を1ピクセル毎に繰り返し、その演算結果を再配置してフィルタ処理画像を得る。

コンボリューション演算は、カーネルサイズ・画像サイズが大きくなると非常に演算負荷が高い。また、汎用プロセッサで画像フィルタ演算を実行する際に問題となるのは、ピクセルデータ読み出し時における複雑なメモリアドレス計算と、大量の冗長なメモリアクセスである。メモリからデータを読み出すためには、ピクセルの画像内の座標から一意に決まるメモリ上の物理アドレスを逐次計算しなければならない。また、カーネルを画像全体に対してスキャンしている間と同じピクセルデータが何度も読み出される。つまり、画像フィルタ演算は演算負荷が高いだけでなく、演算効率が悪く消費電力の大きい処理である。

フィルタ演算を高速に実行するための専用 VLSI の研究はアナログ回路技術[1]、デジタル回路技術[2,3]の両面から進められている。その中で、ピクセル毎に演算回路を設けて全ピクセルを並列に処理する専用プロセッサがいくつか開発されている。しかし、各ピクセルの演算回路を相互に接続するための配線が膨大かつ複雑になることが欠点である。また、ピクセル毎の演算ユニットは、画像と同じく2次元的に接続しなくてはならず、物理的なレイアウトに柔軟性がない。従って、プロセッサに実装できる演算機能や、取り扱うことのできる画像サイズに大きな制約が生じてしまう。

こうした背景をふまえ、本研究では様々なカーネルによるコンボリューション演算を1クロックサイクルで実行する画像フィルタ演算プロセッサの開発を行った。本プロセッサは、物理的レイアウトに柔軟性を持ち、より高度アルゴリズムを実現する画像処理プロセッサのマクロモジュールとして動作させることができる。提案手法の性能を検証するために、5層メタル0.18- μm CMOS プロセスにてプロトタイププロセッサの試作を行い、いくつかの典型的なカーネルに対する画像フィルタ演算の処理の結果を示した。

2. アーキテクチャ

2.1. システム構成

図1に画像フィルタ演算プロセッサの基本アーキテクチャのブロック図を示す。プロセッサ内には、独立で動作する4つの Processing Element (PE) が配置される。PE は画像データを蓄える SRAM モジュールと、読み出した画像データに対しコンボリューション演算を行うローカルコンボリューション演算器 (LCU) から構成される。LCU は Logic-in-Memory[4-6]のコンセプトに基づき、SRAM モジュールのセンスアンプ回路に直結する。フィルタ演算に必要なカーネルは、カーネル制御器(Kernel Controller)に保存され、コンボリューション演算時にそれぞれの LCU へ分配・伝送される。グローバルコンボリューション演算器は4つの LCU の出力を総和を計算しコンボリューション演算の結果を出力する。プロセッサの制御するために入力された命令は、Instruction Decoder によって、各ユニットを制御するためのコードにデコードされる。

2.2. Quaternary Tile Mapping 法

図2に本研究で提案する Quaternary Tile Mapping (QTM)法を示す。まず図2(a)のように、フィルタ演算対象の画像を4x4ピクセルのピクセルブロックに分割し、隣り合うブロックがそれぞれ別のグループになるように、 $a \cdot b \cdot c \cdot d$ の4つのグループに分ける。そしてaグループに属するブロックは全て PE a、bグループは全て PE b、というように、各グループのピクセルデータを対応する PE 内の SRAM モジュールに書き込む。この時、一つのブロック(4x4ピクセル)に含まれる全データは SRAM モジュールの同じ行アドレスに書き込む。

コンボリューション演算を行うときは、図2(b)のように4つの PE を同時に動作させ、カーネルが存在する領域に対応する行アドレスにそれぞれアクセスする。即ち、4x4ピクセルブロックを4つを合わせた8x8ピクセルの領域が LCU に読み出される。4つのピクセルブロックを常に同時に読

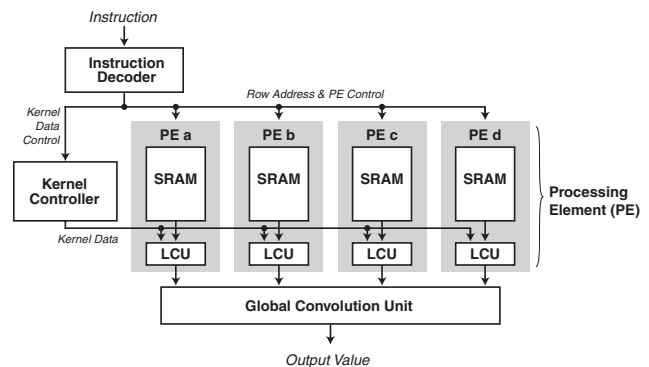


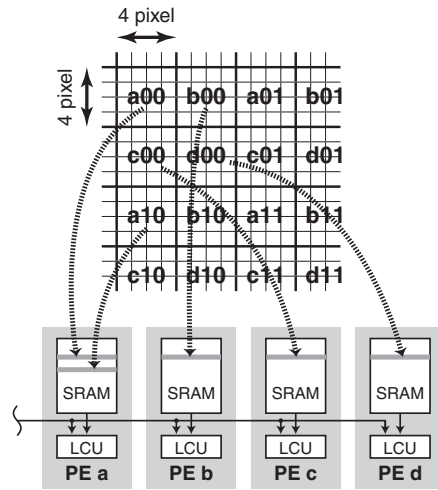
図1: 画像フィルタプロセッサの基本アーキテクチャ

み出すことで、カーネルサイズが 5x5 以下であれば、画像のいかなる部分画像を対象としたコンボリューション演算でも、必要なデータを全て一度のアドレス指定で読み出すことが可能である。5x5 以上の場合も、複数のカーネルを交換しながら複数サイクルに分けて演算することで処理する。また、一度 LCU にダウンロードされたデータは LCU 内のレジスタにキャッシュされるため、カーネルがこの 8x8 ピクセルの領域に位置する限り、SRAM モジュールに冗長なアクセスを発生させることなくコンボリューション演算を繰り返すことができる。

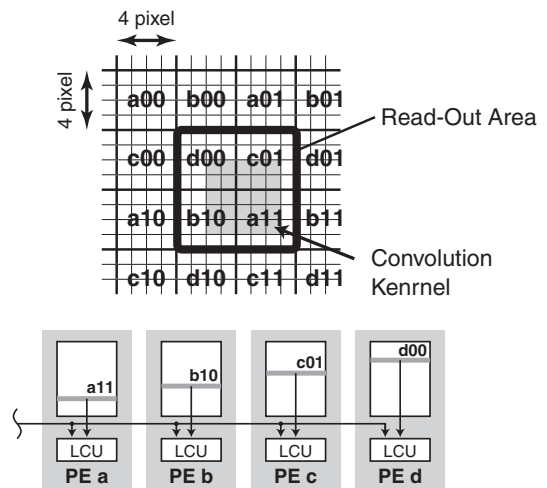
このようにして、画像データのアクセスに必要であったピクセル座標値の演算を、単純なアドレス変換へと簡略化すると同時に、冗長なメモリアccessの排除した。また、各ピクセルブロックのデータは、2次元から1次元に展開してSRAMに保存されるため、従来のピクセル並列演算プロセッサに比べ、物理的なレイアウトの自由度が高い。

2.3. カーネルデータの制御

カーネルデータの制御法を、図3に示す。本プロセッサで1サイクルで演算できるカーネルの大きさは最小 2x2 ピクセル~最大 8x8 ピクセルである。カーネルデータはカーネル制御器(Kernel Controller: KC)内のデータレジスタに、8x8 サイズのマップデータとして保存する。カーネルが 8x8 ピクセルより小さい場合は、使用しない領域は No-Operation(NOP) を表す 0 で埋められる。演算時には、4つの LCU に読み出された 8x8 ピクセルの部分画像の演算対象部分に位置するように、マップデータを x、y 両方向にブロックシフトする。そして 4x4 サイズの小さなマップ4枚に分割し、それぞれ対応する LCU へ伝送する。このように、



(a) 画像データのメモリ回路への分配



(b) フィルタ演算実行時のメモリアccess

図2: Quaternary Tile Mapping 法

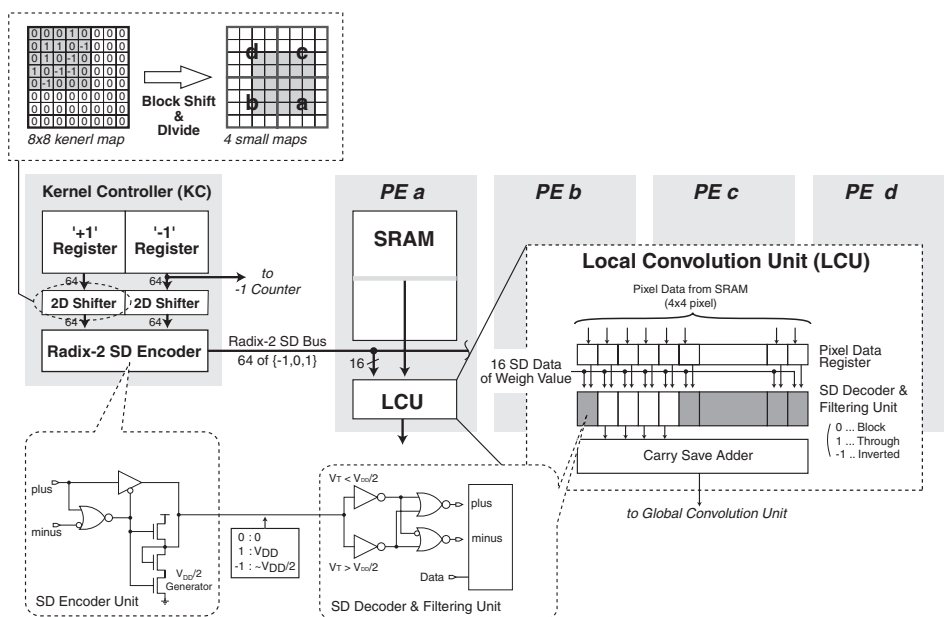


図3: Kernel Controller 及び Local Convolution Unit における処理

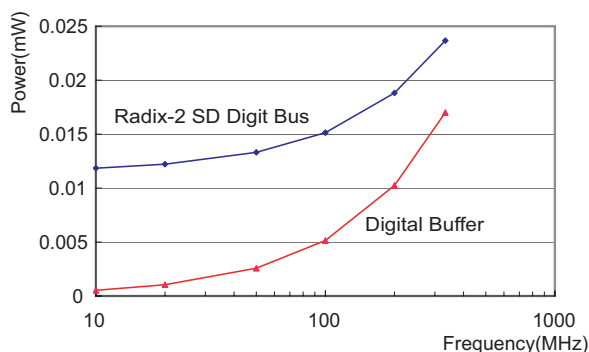


図 4：データ転送における消費電力のシミュレーション

ピクセルデータの座標値の演算をカーネルデータのシフト演算置き換えて処理する。

本プロセッサでは、カーネル内の係数は 0、+1、-1 の 3 種類を扱うことができる。KC 内では、カーネルデータは使用するカーネルの種類に関係なく +1 と -1 が独立し二つの 8x8 の係数マップとしてデータレジスタに蓄えられる。例えば +1 のマップの場合、係数 +1 が割り当てられているところにビット 1 が割り当てられて、そのほかの場所には NOP としてビット 0 が割り当てられる。

KC 内で独立し処理されていた二つの 8x8 マップを、LCU へ伝搬する際は、本研究で新たに提案する Radix-2 Signed-Digit(SD) Data Transfer 方式を用いる。通常のデジタルデータバスでは、 V_{DD} に論理 1 を、 V_{SS} に論理 0 を割り当てて、Radix-2 SD Data Transfer は、 V_{DD} 、 V_{SS} に加えて $V_{DD}/2$ に “-1” を割り当て、1 本のバスラインで 3 つの状態の伝送を実現する。カーネルデータの中では、+1、0 に比べ -1 の出現頻度は低いので、-1 を伝送する専用のデジタルバスを用意する必要は低い。従って、この Radix-2 SD Data Transfer を使うことで、カーネルデータ伝送におけるバスの配線を効果的に利用することができ、チップのレイアウト面積を抑えることができる。

Radix-2 SD Data Transfer のエンコーダとデコーダの回路図を図 3 に示す。エンコーダは、通常のデジタルデータを伝送するためのバッファと、ソースフォロワを応用した $V_{DD}/2$ -Generator から構成される。デジタルバッファでデータを伝送する場合、負荷を目的の電圧まで駆動した後は電流が流れず、配線容量を充放電するためだけに電力が消費される。一方、 $V_{DD}/2$ Generator は配線を駆動した後も Generator 内を貫通電流が流れてしまう。図 4 に、“+1” と “-1” の重み係数が均等に配置されたカーネルデータ(後述の図 8(C)のカーネル)を、通常のデジタルバッファと本プロセッサで用いた Radix-2 SD Data Transfer で伝達した時の消費電力のシミュレーション結果を示す。配線負荷は、プロセッサ内の一部の配線をモデルで近似したモデルを用いた。100MHz 以下の低い周波数ではデジタルバッファに比

べ Radix-2 SD Data Transfer は消費電力が大きい。しかし、100MHz 以上の高周波数においては配線駆動にて消費される電力が支配的になるため、消費電力の差は小さくなる。本プロセッサで設計した Radix-2 SD Data Transfer 方式のデータバスは最高 300MHz までの伝送能力を持つ。しかし、動作周波数に合わせて回路パラメータの最適化を行い、配線駆動に必要な時間のマージンを少なくすることで消費電力を低減できると考えられる。

Radix-2 SD Data Decoder は、論理閾値が $V_{DD}/2$ より大きい値に調節したインバータと、小さい値に調節したインバータを用いる。入力が +1(V_{DD})、0(V_{SS})の時は二つのインバータは同じ値を出力するが、-1($V_{DD}/2$)が入力されると閾値が大きいインバータは論理 1、小さいインバータは論理 0 を出力する。これを、NOR 回路で検出することで、Radix-2 SD Data から、元のデジタル信号へ復号する。

2.4. Local Convolution Unit

ローカルコンボリューション演算器(LCU)は、読み出したピクセルデータをキャッシュするレジスタ、Radix-2 SD Decoder、フィルタリングユニット、キャリアセーブ加算器から構成され、4x4 領域のピクセルデータ個々に対する係数演算とその結果の加算を行う。LCU は SRAM モジュールのセンス・アンプに直結する形で実装し、4x4 領域のピクセルデータを 1 サイクルで全て同時に読み出して並列に処理する。

読み出されたピクセルデータは、カーネル係数に基づいて次の処理を行う。まず係数 0 の場合は、次段の加算に影響しないように、ピクセルデータも 0 に変換する。一方、係数が +1 の場合はピクセルデータそのままの値が、係数 -1 の場合はビット反転した値が、それぞれ次段の加算器に入力される。最後にキャリアセーブ加算器を用いて、処理されたピクセルデータの総和が演算される。

並列に演算された 4 つの LCU の出力は、グローバルコンボリューション演算器にて総和が計算され、演算結果として出力される。このとき、カーネル係数の -1 の総和を LCU とは別に計算しておき、最後にグローバルコンボリューション演算器で足し合わせる。プロセッサ内の加算器は全て一つのキャリアセーブ加算器として構成され、最小限の回路構成で +1、0、-1 という 3 種の係数を用いた 8x8 のコンボリューション演算を効率よく実行する。

3. 測定結果

3.1. 試作プロセッサ

プロトタイププロセッサを5層メタル0.18- μ mCOS プロセスを用いて設計・試作を行った。試作したプロセッサの、チップ写真を図5に、主な仕様を表1に示す。設計は集積度を上げるために、自動配置配線ツールを用いず全てマニュアルレイアウトで行った。配線遅延・信号スキューを最小にするために、4つのPEは更に小さなユニットに分割して対称的に配置した。このように、実装する領域に適合するように柔軟な設計を行うことができるのも、提案手法の特徴である。プロセッサ内にカーネル用のメモリを実装し、複数のカーネルデータを書き込み、カーネルを切り替えながら画像フィルタ演算を実行可能である。

図6にプロセッサの動作波形を示す。これは画像フィルタ演算に、図7(c)で示す縦方向エッジ抽出フィルタを用い、電源電圧1.8V、クロック周波数50MHzで動作させた時の波形である。入力命令のデコードに6ns、コンボリューション演算の実行に11ns、合計で17nsが処理に必要な時間であった。本プロセッサは概念の実証を主な目的としているため、命令のデコードから演算処理の出力まで全てを1サイクルで処理するように設計を行った。従って、処理をパイプラインで分割することで更に動作周波数を高めることができると考えられる。消費電力は、コンボリューション演算に使用するカーネルの種類により変動するが、典型的な5x5のカーネルに対して、平均で180mWであった。

3.2. 画像フィルタ演算の実行結果

試作したプロセッサで、画像フィルタ処理を行った結果を図7に示す。128x128ピクセルの原画像と、4種類のカーネルをプロセッサに書き込み、それぞれコンボリューション演算を行った値を再配置したものである。なお試作プロセッサには除算機能を実装していないため、外部に設けた汎用プロセッサにより除算を実行した。3x3平滑化フィルタ、5x5平滑化フィルタ、5x5縦方向エッジ抽出フィルタ(順に(a)、(b)、(c)に対応)に対して、1クロックでコンボリューション演算を実行することを確認した。(d)にガウシアンフィルタカーネルの処理結果を示す。ガウシアンフィルタに必要な係数は他フィルタに比べ複雑であるため、コンボリューション演算は図中に示すような3つの単純なカーネルの重ね合わせによって実現した。つまりガウシアンフィルタの実行には3サイクル必要である。しかし、本プロセッサを用いることで最も処理負荷の高いピクセルデータの読み出し処理を高速化し冗長なメモリアクセスを省くことができるため、通常のMPUを用いるシステムに比べ高速化・低消費電力化が見込まれる。

図8に、本プロセッサと汎用プロセッサとの画像フィル

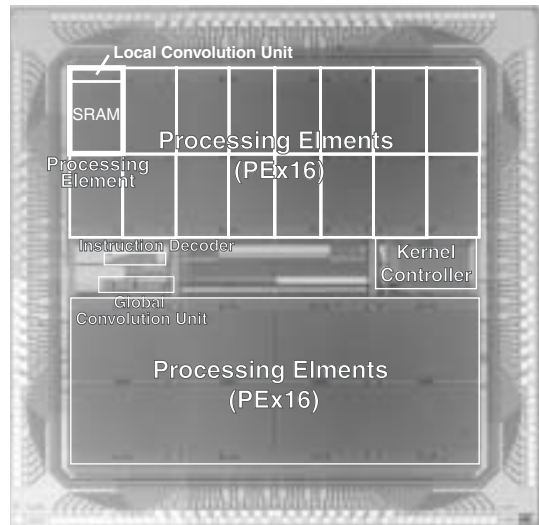


図5：試作プロセッサのチップ写真

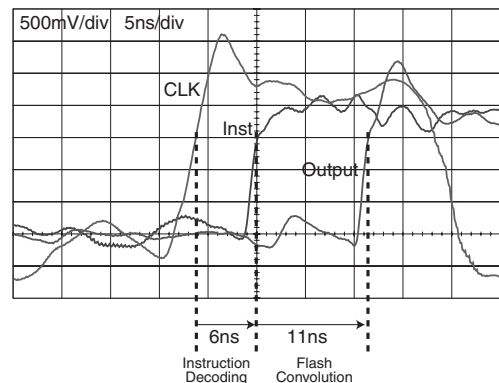


図6：試作プロセッサの動作波形

表1：試作プロセッサの主な仕様

Process Technology	0.18 μ m CMOS, 5-Metal
Core Size	4.5 mm x 4.6 mm
Transistor Count	4,000k (SRAM: 3,500k Logic: 500k)
Supply Voltage	1.8V
Operating Frequency	50MHz
Power Consumption	180mW (for Typical Kernels)
Memory Size	64k Bytes for Image, 2k Bytes for Kernel Data
Kernel Size	2x2 ~ 8x8 variable

タ処理実行時間の比較結果を示す。それぞれ、256x256ピクセルの画像をメモリ上にロードし、いくつかのカーネルを用いて画像フィルタ演算を実行し、その必要時間を測定

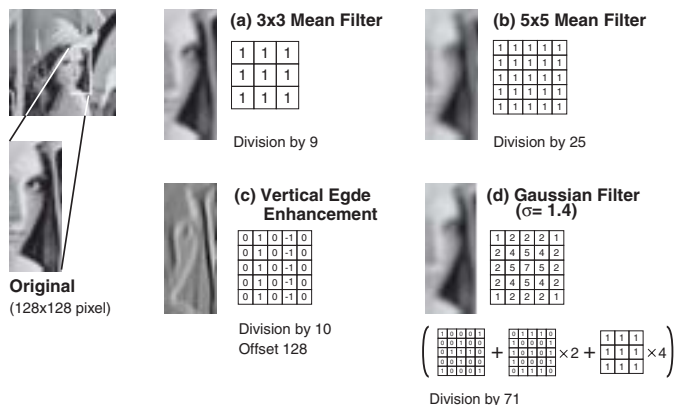


図 7：画像フィルタ演算の結果

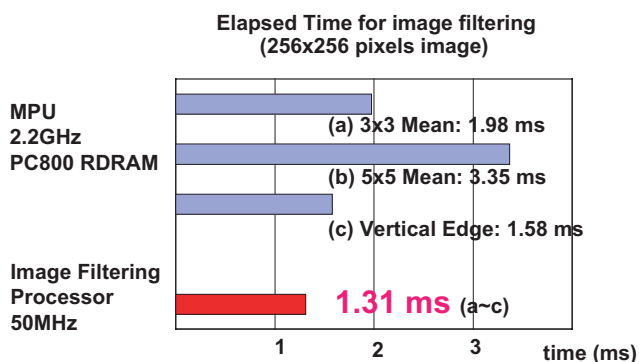


図 8：汎用プロセッサとの性能比較

した。Pentium4 2.2GHz、メモリ 1G Bytes という構成の汎用コンピュータ上にて、C 言語を用いて作成した画像フィルタプログラムを実行し計測を行った。なおグラフに示した数値は、図 8 に示した(a)~(c)のカーネルを用いて 1 万回フィルタ演算を繰り返した場合の 1 回当たりの実行時間の平均値である。

汎用プロセッサによるフィルタ処理の実行時間は、1.5 ~ 3ms 程度であった。これは、各カーネル内に配置された 0 以外の重み係数の数に比例して変化している。一方、画像フィルタ演算プロセッサは、1 クロックサイクルにつき 1 回のコンボリューション演算が実行されるため、処理の実行時間は、カーネルによらず 1.31ms である。プロセッサで消費される電力は、動作周波数に比例する。従って、本プロセッサを用いることにより、汎用プロセッサに比べ 1/20 程度の消費電力で、同程度以上の速度の画像フィルタ演算を実現できることを示した。

4. まとめ

様々なカーネルによるコンボリューション演算を 1 クロックサイクルで実行する画像フィルタ演算プロセッサの開発

を行った。画像データを 4x4 のピクセルブロックに分割して、隣り合うブロックが異なる PE によって処理されるようにメモリマッピングを行う Quaternary Tile Mapping 法を開発した。Quaternary Tile Mapping 法により、ピクセルデータを読み出す際に必要であった複雑なアドレス計算を単純なカーネルのシフト演算に置き換え、冗長なメモリアccessを排除することで、高速・低消費電力な画像フィルタリング処理を実現した。0.18- μ m CMOS プロセスにてプロトタイププロセッサを設計し、電源電圧 1.8V、クロック周波数 50MHz、消費電力 180mW で動作することを確認した。試作プロセッサを用いて典型的な 4 種類のカーネルについての画像フィルタ演算の結果を示し、汎用プロセッサ比べ、約 1/20 の消費電力で同速度の画像フィルタ処理を実現できることを示した。

参考文献

- [1] D. A. Martin, H. S. Lee, and I. Masaki, "A Mixed-Signal Array Processor with Early Vision Applications," IEEE J. Solid-State Circuits, Vol. 33, pp 497-502, March 1998.
- [2] J. C. Gealow and C. G. Sodini, "A Pixel-Parallel Processor Using Logic Pitch-Matched to Dynamic Memory," IEEE J. Solid-State Circuits, Vol. 34, pp. 831-839, June 1999.
- [3] M. Ishikawa, K. Ogawa, T. Komuro, and I. Ishii, "A CMOS Vision Chip with SIMD Processing Element Array for 1ms Image Processing," Dig. Tech. Papers of 1999 IEEE Int. Solid-State Circuits Conf., TP12.2, pp. 206-207, San Francisco, U.S.A., February 1999.
- [4] H. S. Stone, "A Logic-in-Memory Computer," IEEE Trans. on Computers, Vol. C-19, No.1 pp.73-78, Jan, 1970.
- [5] D. G. Elliott, M. Stumm, W. M. Snelgrove, C. Cojocararu and R. Mckenzie, "Computational RAM: Implementing Processors in Memory," IEEE Design and Test of Computers, vol.16, pp.32-41, No.1, 1999.
- [6] D. Patterson, T. Anderson, N. Cardwell, R. Fromm, K. Keeton, C. Kozyrakis, R. Thomas and C. Yelick, "A Case for Intelligent DRAM: IRAM," in IEEE Micro, vol.17, no.2, pp.34-44, April 1997.

本研究に関する発表

- [1] K. Ito, M. Ogawa, and T. Shibata, "A Variable-Kernel Flash-Convolution Image Filtering Processor," Dig. of Tech. Papers of 2003 IEEE International Solid-State Circuits Conference, pp.470-471, San Francisco, U.S.A., February,2003.
- [2] 伊藤潔人,小川誠,柴田直,"フラッシュコンボリューション型画像フィルタ演算プロセッサ," 電子情報通信学会技術研究報告, (集積回路研究専門委員会(ICD)),2003年5月