

## メモリ共有型マルチSIMDアーキテクチャを有する 高性能ビジョンプロセッサの設計

山口 光太<sup>†</sup> 渡辺 義浩<sup>†</sup> 小室 孝<sup>†</sup> 石川 正俊<sup>†</sup>

<sup>†</sup> 東京大学大学院情報理工学系研究科 〒113-8656 東京都文京区本郷 7-3-1

E-mail: †{Kota\_Yamaguchi, Yoshihiro\_Watanabe, Takashi\_Komuro, Masatoshi\_Ishikawa}@ipc.i.u-tokyo.ac.jp

あらまし 実環境での高速な画像認識のためには、画像の前処理から特徴量抽出に至るまでの多岐に渡る膨大な演算の高速化が課題となっている。これに対し、我々は2次元/1次元SIMD処理モジュールと逐次処理モジュールがメモリを共有するビジョンプロセッサを設計した。提案プロセッサでは各モジュールでのSIMD処理に加えてモジュール間に総和演算/ブロードキャスト機構を設けることで、画像処理に必要な各種演算の効率化を狙っている。また、複数モジュールのメモリ共有によりデータ転送コストの軽減、並列アルゴリズムの実装性向上を実現した。シミュレーションにより様々な画像処理の演算ステップ数を大幅に削減できることが見積もられた。

キーワード 超並列処理, SIMD, 画像認識 LSI

## Design of a High-Performance Vision Processor with Shared-Memory Multi-SIMD Architecture

Kota YAMAGUCHI<sup>†</sup>, Yoshihiro WATANABE<sup>†</sup>, Takashi KOMURO<sup>†</sup>, and Masatoshi ISHIKAWA<sup>†</sup>

<sup>†</sup> Graduate School of Information Science and Technology, The University of Tokyo  
7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8656 Japan

E-mail: †{Kota\_Yamaguchi, Yoshihiro\_Watanabe, Takashi\_Komuro, Masatoshi\_Ishikawa}@ipc.i.u-tokyo.ac.jp

**Abstract** For high speed image recognition in real environment, it is a challenge to accelerate a large amount of calculation for image processing from pre-processing to feature extraction. We designed a vision processor which consists of 2D / 1D parallel SIMD and 0D sequential processor modules that share a memory. The processor has summation and broadcast function between modules, which accelerates complicated operations in image processing. Memory sharing reduces cost for data transferring between modules and simplifies implementation of various parallel algorithms. Simulation results show the processor can perform various image processings with much less operation steps.

**Key words** massively parallel processing, SIMD, image recognition LSI

### 1. はじめに

自動車の運転支援やセキュリティ機器での利用など、実環境での画像認識へのニーズが近年高まっている。実環境での画像認識には高度かつ膨大な演算が必要とされるため、その処理の高速化が課題となっている。画像認識システムのフレームレートや対象認識率の向上のためには高速な処理を実現するプロセッサが必要である。

これまでも画像認識を想定したプロセッサはいくつか研究されてきた。NECが開発したIMAPアーキテクチャは128個のPE(Processing Element)を直線状に並べ、画像の列単位で並列処理を可能としている[1]。また、東芝は3つのVLIWプ

ロセッサモジュールを搭載した画像認識プロセッサViscontiを開発し、データ流および命令レベルでの並列処理を実現している[2]。

これらのプロセッサは高度な処理を実装し、車載用途などで一定の成果を挙げている。しかし、ロボットビジョンなど実環境でのリアルタイムな画像認識へ応用するには現状以上の高い性能が求められる。空間、物体や人物の認識のためには、多岐に渡る画像処理の更なる高速化が必要と考えられる。

多量のデータに対し繰り返し同一内容の演算を行うような単純な画像処理においては、SIMD(Single Instruction Multiple Data)型並列プロセッサの有用性がこれまで示されてきた。特に画素単位でPEを置いた超並列SIMD型プロセッサでは逐次

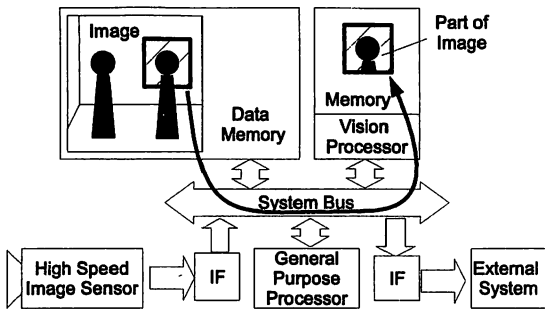


図1 高速画像認識システム

処理に比べて大幅に処理効率を向上できることが知られている。東京大学の石川らの研究グループはイメージセンサの画素毎に PE を配置したビジョンチップを開発し、高速な各種フィルタ処理や、対象の位置、面積などの特徴量抽出を実装している [3]。これにより従来不可能であった実時間での視覚計測、視覚フィードバック制御などが実現されている。Gealow らは画素並列の PE を用いた低コストな高速画像処理を目指している [4]。また、Gayles らは 49152 個の PE を使い、画像をはじめとして各種信号処理への応用を試みている [5]。

しかし、これらの超並列プロセッサを画像認識に用いた例はこれまで多くは見られなかった。画像認識で行われる処理は多岐に渡り、その複雑な演算を超並列プロセッサに実装するのが困難であったこと、また超並列プロセッサへのデータの入出力がボトルネックとなり実装のメリットを享受しにくいことなどがその理由と考えられる。ポリゴン図形のマッチングを超並列に行う試みなどは見られたが [6]、実環境での画像認識へ応用するには実装できる処理の制約が大きい。画像認識を行うためには汎用の逐次処理型プロセッサや、並列度の小さな SIMD 型プロセッサを用いることが多かった。

画像認識の高速化という課題に対し、我々は超並列 SIMD 型アーキテクチャの高い処理能力を様々な画像処理に利用可能とするビジョンプロセッサの設計を行った。このプロセッサは複数の並列度の異なる SIMD 型プロセッサモジュールが一つのメモリを共有する。このマルチ SIMD アーキテクチャにより従来の超並列プロセッサで問題となる複雑データ流処理の強化、データ転送コストの削減が可能となる。

提案するビジョンプロセッサは図 1 に示すような高速画像認識システムに用いることを想定している。画像認識では必ずしも画像全体に処理を適用するのではなく、局所的に切り出された画像に処理を適用する場合が多い。この局所的な画像に対して超並列処理を行うことで、高速な画像認識が実現されると考えられる。

## 2. アーキテクチャの設計

### 2.1 画像認識に必要な処理

空間、物体、顔など、画像認識には対象に応じて様々な手法が存在する。その多くは以下のような画像処理を組み合わせることで実現される。

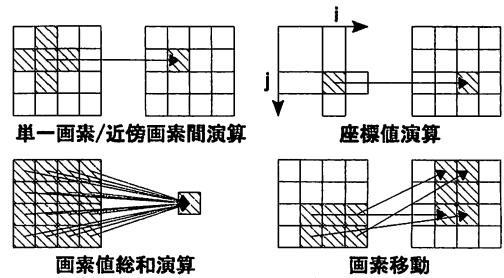


図2 画像認識で必要とされる演算

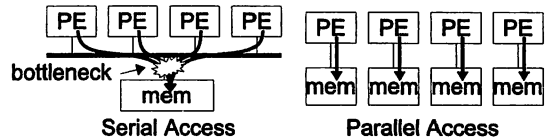


図3 並列プロセッサによるメモリアクセス

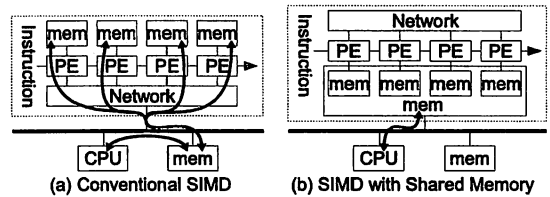


図4 SIMD 型プロセッサにおけるデータ転送

- 前処理: 2 値化、2D フィルタリングなど
- 中間処理: スカラー特徴量抽出、図形描画、アフィン変換、直交変換など

これらの処理によって抽出された特徴量を基に識別などの高次処理を行う。実環境での画像認識のためには、その解空間の探索にかかる膨大な演算を効率化する必要があった。

### 2.2 従来の超並列 SIMD 型処理の問題

前述の画像処理で多用される演算のうち代表的なものを図 2 に示す。例えば図形描画は座標値演算と単一画素演算を組み合わせることで実現される。

画素並列に PE を配置した従来の超並列 SIMD 型プロセッサでは単一画素/近傍画素間演算を高速に実行することができる。しかしその他の複雑な演算は、PE 間の結合が限られた超並列 SIMD 型プロセッサでは扱うことが難しい。PE 間の結合を可変とすることで複雑なデータ流を扱う試みはあるものの [5] [7]、高速な画像認識のためには更に処理に自由度を持たせたアーキテクチャが必要と考えられる。

また、SIMD 型プロセッサは PE 毎にメモリを分散して配置するため、図 3 に示すように共有メモリ型の並列プロセッサにおけるメモリボトルネックを回避することができる。これにより超並列 SIMD 型プロセッサは並列度に比例した性能向上が期待できた。

しかし、一方でデータの転送方法が問題となっていた。図 4(a) に示すように、超並列 SIMD 型プロセッサが分散して持

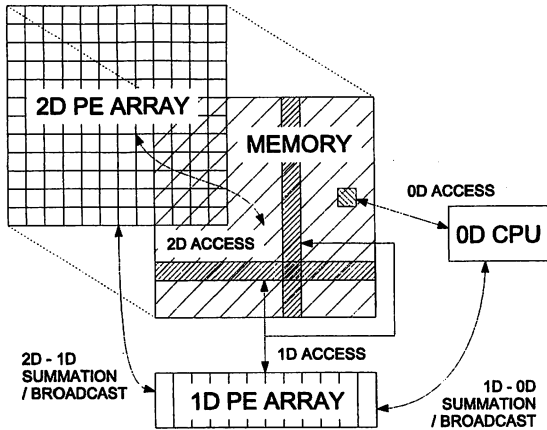


図5 メモリ共有型マルチSIMDアーキテクチャ

つメモリのデータを外部へ入出力する場合、逐一PEを介したデータの転送を行う必要があった。このデータ転送にかかるコストがシステム全体のスループットを押し下げている要因になるとともに、超並列SIMD型プロセッサを利用しにくいものになっていた。

複数プロセッサ間でのデータ転送コストは、メモリの共有により抑えることができる。したがって図4(b)に示すようにコントローラと並列プロセッサとがメモリを共有する構成を用いることで超並列SIMD型プロセッサのデータ転送コストを軽減できるものと考えられる。

### 2.3 マルチSIMDアーキテクチャ

従来の超並列SIMD型における問題に対し、我々は図5に示すマルチSIMDアーキテクチャを設計した。このアーキテクチャはメッシュ状にPEを結合した2次元SIMDモジュール、直線状にPEを並べた1次元SIMDモジュール、逐次処理を行う0次元モジュールがメモリを共有する構造を持つ。共有メモリは2次元SIMDモジュールのPEの数だけバンクを持つもので、その詳細は後述する。2次元/1次元SIMDモジュール間と1次元SIMD/0次元モジュール間にはそれぞれ図6に示す総和演算/ブロードキャストの機構を設けた。0次元モジュールには汎用のプロセッサを用いることを想定するが、SIMDモジュールのコントローラとしての機能も持たせるものとする。以下、マルチSIMDアーキテクチャの特長について述べる。

#### a) 2次元/1次元SIMDモジュール

2次元SIMDモジュールはメッシュ状に並べたPEにより画素並列処理を行う。各PEは上下左右の4近傍のPEと通信を行うことが可能で、単一面素演算や近傍画素間演算を効率的に実行することができる。図7に2次元SIMDモジュールにより1ステップで可能な演算の具体例を示す。PEは4近傍や共有メモリからのデータを選択するマルチプレクサとALU、演算のキャリヤやボローを保持するレジスタなどから構成する。ただし、画素数だけPEを用意することで増大する回路規模との兼ね合いから、扱えるビット幅や演算機能は必要最小限に抑える必要がある。

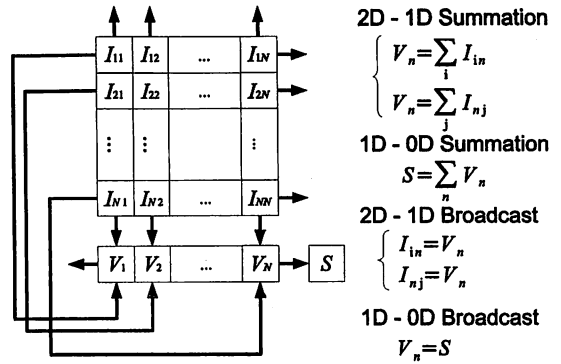


図6 総和演算、ブロードキャスト

#### <Pixel Parallel Operation>

$$\begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 2 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 2 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 3 & 2 & 1 \\ 1 & 2 & 3 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

#### <Neighbor Communication>

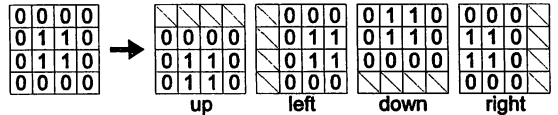


図7 2次元SIMDモジュールを用いた演算の例

1次元SIMDモジュールは画像の幅だけPEを直線状に結合したもので、1次元状のデータを並列に演算することができる。また2次元SIMDモジュール同様に左右の近隣PE間で通信可能であるものとする。2次元SIMDモジュールではハードウェアコストの問題から個々のPEの機能を抑えたものにする必要があったが、1次元SIMDではそれと比較して高機能なPEを用いることが可能である。例えば演算器の扱えるビット幅を2次元SIMDモジュールより大きくしたり、乗算器の搭載などが考えられる。それにより座標値など画像の列/行単位で共通となるデータを効率的に処理することが可能となる。

#### b) 総和/ブロードキャスト機構

総和演算機構は2次元データから1次元データ、1次元データからスカラー値を、総和をとることにより計算するものであり、ブロードキャスト機構はスカラー値を1次元データに、1次元データを2次元データに展開する操作を行うものである。2次元/1次元SIMDモジュールを用いた総和/ブロードキャストの演算例を図8に示す。

総和演算は例えば図9のようにPEの持つALUを直列に接続することで実現可能である[8]。一方、ブロードキャストは共有メモリの複数のバンクに対して同時に書き込みを許可することで実現可能である。詳細は後述する。

総和演算により画像全体からの特徴量の抽出計算、ブロードキャストにより座標値を用いた演算などの効率化が期待できる。また、組み合わせることで図10に示す線形変換も効率化することができる。1次元SIMDモジュールからベクト

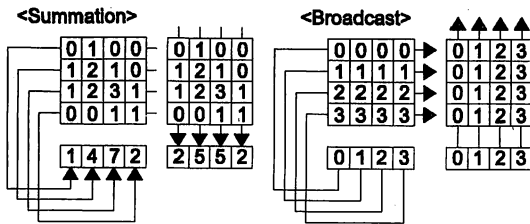


図 8 2次元/1次元 SIMD 間の総和/ブロードキャストの例

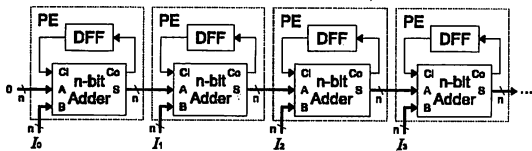


図 9 n ビットシリアル総和演算回路

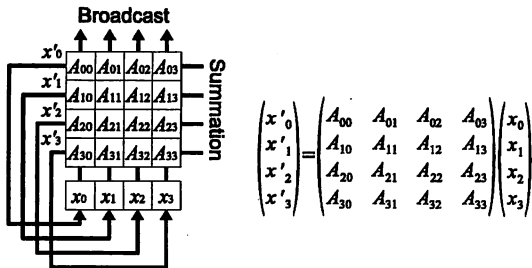


図 10 並列線形変換

ル  $\alpha$  を列ブロードキャストし、2次元 SIMD によってこれに行列  $A$  を乗じる。続いて行方向に総和演算を行うことでベクトル  $\alpha'$  を得ることができる。これを用いることで1次元 SIMD モジュールの任意の PE 間でのデータ交換などが可能である。

### 3. アーキテクチャ評価

マルチ SIMD アーキテクチャと従来の SIMD 型アーキテクチャ、逐次処理アーキテクチャそれぞれにおける画像処理の演算ステップ数をシミュレーションにより見積もり、比較を行った。

#### 3.1 見積もり条件

逐次処理 (0D)、列並列 SIMD (1D)、画素並列 SIMD (2D)、マルチ SIMD (M-SIMD) の4種類のアーキテクチャで6種類の画像処理を実行するのにかかる演算ステップ数を見積もった。シミュレーションした処理は2値化、エッジ検出、円の描画、モーメント特徴量抽出、バイナリ画像の回転、1次元 FFT の6つで、いずれも画像認識で重要とされるものである。

2D の PE は1ビット、1D の PE と 0D のプロセッサは任意のビット幅の整数を1ステップで処理できるものとした。M-SIMD が扱えるデータ幅はこれらを組み合わせたものとした。M-SIMD の総和演算、ブロードキャストはそれぞれ1ステップで実行できるものとした。画像サイズは  $64 \times 64$  と  $256 \times 256$  の2種類について見積もりを行った。

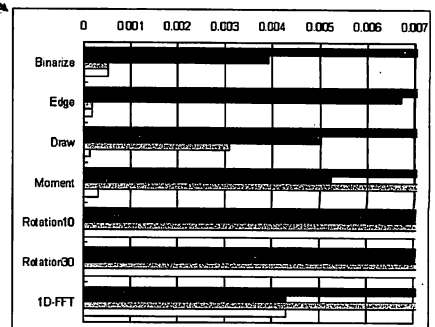
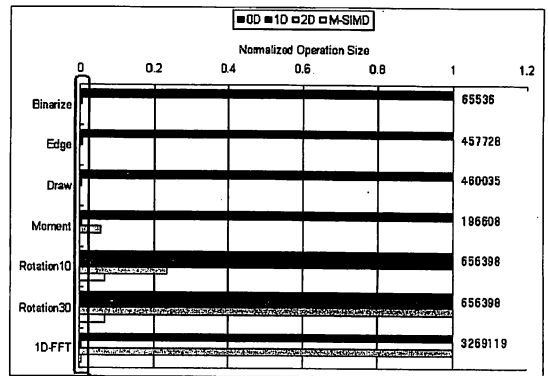


図 11 演算回数を見積もり

各アーキテクチャで6種類の画像処理にかかる演算ステップ数のオーダを表1に示す。N は画像の縦横の幅、M は2次元の PE で近傍通信を用いて画像を移動させる範囲  $M \times M$  を表す値で、その大きさは画像サイズや回転の角度、中心座標に依存する。

### 3.2 結果

見積もりの結果を図11に示す。M-SIMD は0D に比べ演算ステップ数をおよそ1/10 から1/3000 程度に削減可能であることが確認された。特に1D や2D 単独で達成可能な処理効率よりも、1次元と2次元の SIMD モジュールを組み合わせた M-SIMD の処理効率のほうが高いことが確認された。なお、サイズ  $64 \times 64$  の画像での見積もりについてもほぼ同様な傾向が見られた。

これらのことから、画像認識に必要なとされる複雑な画像処理にマルチ SIMD アーキテクチャは高い演算能力を発揮すると見える。

### 4. アーキテクチャ考察

マルチ SIMD アーキテクチャに関して、次の3つの観点から考察を行った。

#### a) ハードウェア量

シミュレーションではマルチ SIMD アーキテクチャにより種々の処理を効率的に実行可能であることが示された。しかし複数の SIMD モジュールを搭載することにより増大する回路面積などのコストも考慮しなくてはならない。

表 1 演算量のオーダー

	0D	1D	2D	M-SIMD
2 値化	$O(N^2)$	$O(N)$	$O(1)$	$O(1)$
エッジ検出	$O(N^2)$	$O(N)$	$O(1)$	$O(1)$
円の描画	$O(N^2)$	$O(N)$	$O(1)$	$O(1)$
モーメント特徴量抽出	$O(N^2)$	$O(N)$	$O(N)$	$O(1)$
バイナリ画像の回転	$O(N^2)$	$O(N^2)$	$O(M^2)$	$O(N)$ or $O(M^2)$
1 次元 FFT	$O(N^2 \log N)$	$O(N \log N)$	$O(N^2 \log N)$	$O(N \log N)$

一般に  $n$  ビットの乗算器のハードウェアコストを  $n^2$  とする。また、マルチ SIMD アーキテクチャでは逐次処理モジュールに  $n_0$  ビットのプロセッサ、1 次元 SIMD モジュールに  $n_1$  ビットの PE を  $N$  個、2 次元 SIMD モジュールに  $n_2$  ビットの PE を  $N \times N$  個並べるとする。このときマルチ SIMD アーキテクチャのコスト  $C_{MS}$  は  $C_{MS} = n_0^2 + n_1^2 N + n_2^2 N^2$  と考えることができる。例えば  $n_0 = 32$ 、 $n_1 = 8$ 、 $n_2 = 1$ 、 $N = 256$  とすれば、 $C_{MS} = 82944$  となる。従来の超並列 SIMD 型プロセッサの PE も同様に 1 ビット幅の演算器をもつとすれば、そのハードウェアコストは 65536 と見積もられる。従ってマルチ SIMD アーキテクチャは従来の超並列 SIMD 型プロセッサに対しおよそ 1.27 倍程度のコスト付加のみで実装可能であると考えられる。

一方、汎用の逐次プロセッサと比較した場合、例えば 32 ビットのプロセッサのハードウェアコストは 1024 と見積もられるため、マルチ SIMD アーキテクチャはそれに比べておよそ 81 倍ものコスト増となってしまいます。しかし超並列化による演算回数削減効果はそれだけのコストに見合ったものが期待できる。

また、汎用の逐次プロセッサを用いたシステムにおいては演算器よりもむしろメモリ回路の規模がハードウェアコストに対して支配的となる。メモリ回路も含めて回路規模を考慮した場合、共有メモリを持つマルチ SIMD アーキテクチャは十分現実的なハードウェアコストで実現可能であると考えられる。

b) メモリアクセス

メモリアクセスの速度は処理全体の実行速度に大きく影響するため、画像処理に限らずこれを高速化することは重要である。マルチ SIMD アーキテクチャの SIMD モジュールは高い並列度でのメモリアクセスを行うため、逐次処理に比べてアクセスバンド幅を大幅に増大できる。これにより逐次処理型のシステムで問題となるメモリアクセスのボトルネックの解消が期待できる。

c) ユーザビリティ

従来の超並列 SIMD 型プロセッサでは逐次型プロセッサとは異なる並列データ型のために、独特のアルゴリズム開発をユーザに強いることとなっていた。マルチ SIMD アーキテクチャは複数のモジュールのメモリ共有により、同一データを 0 次元/1 次元/2 次元で選択的に並列処理することができる。モジュール間の転送命令が不要となるだけでなく、ライブラリ化が容易になるなどの利点がある。また、SIMD モジュールに実装できない処理があったとしても逐次処理モジュールを用いることで実装が可能である。

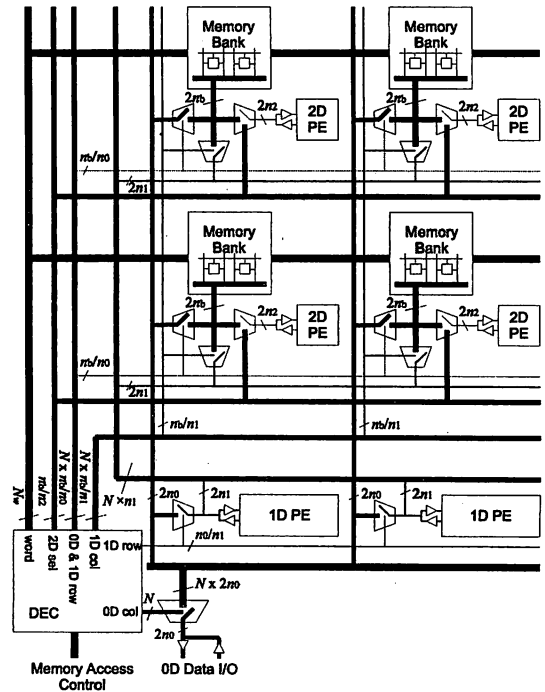


図 12 共有メモリ回路

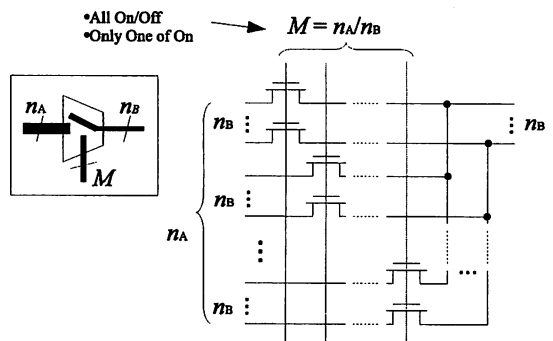


図 13 3 状態セレクタ

5. 共有メモリ回路

0 次元/1 次元/2 次元の各モジュールがアクセスする共有メモリは、それぞれの並列度のアクセスポートを持つ必要がある。汎用のメモリでは利用可能なポート数が限られるため、専用の

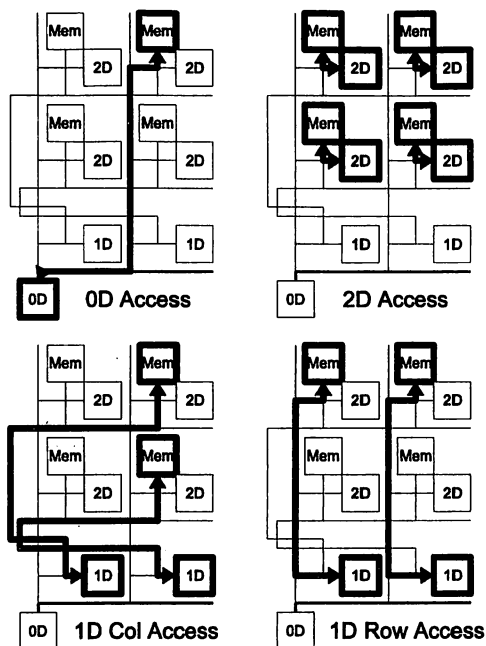


図 14 各モジュールのメモリアクセス経路

共有メモリ回路の設計を行った。図 12 にその回路図を示す。

図中の  $n_0, n_1, n_2$  はそれぞれ 0 次元/1 次元/2 次元の PE が扱うビット幅、 $n_b$  はメモリバンクの持つビット線の数を表す。また  $N$  は画像の幅、 $N_w$  はワード線の数である。台形で囲んだスイッチは、接続の on/off の切り替えスイッチとビット幅の異なる配線間の接続の選択に用いるセクタを兼ねる 3 状態セクタである。図 13 に示すように通常のセクタと同様の構成であるが、制御信号は 1 つだけ on、または全て on/off を入力して用いる。

図 14 に示すように、このスイッチを用いることでメモリアクセスを行うモジュールに応じてデータ線の経路を切り替えてバンクを共有することができる。例えばワード線と任意の列の 1Dcol 制御線をそれぞれ 1 本だけ on にし、他の制御線を全て off にすることで 1 次元 SIMD モジュールによる縦 1 列のメモリバンクへのアクセス経路に切り替わる。

また、図 15 のように複数のメモリバンクに対して同一データを書き込むことでブロードキャストが実現する。例えば 0 次元モジュールからの書き込み時に 0D col 制御線を全て on にし、ワード線と任意の行の 0D & 1D row 制御線を 1 つだけ on にすることで、0 次元から 1 次元へのブロードキャスト書き込みが可能となる。

メモリバンクは双対ビット線とメモリセルを多数並べた汎用の SRAM と同じ構造である。例えばバンク 1 つあたりの記憶容量 512bit、画素数  $256 \times 256$  で構成にした場合、マルチ SIMD アーキテクチャ全体での記憶容量はおおよそ 4MB となる。このときメモリセルがトランジスタ数の大部分を占めることになるため、同容量の汎用 SRAM に数%のトランジスタを付加した程度の回路規模で複数 SIMD モジュールでの共有メモリが

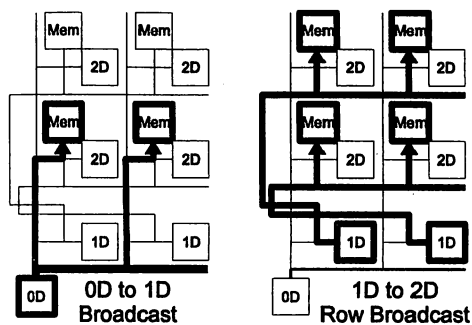


図 15 ブロードキャスト書き込み

実現できると考えられる。

## 6. むすび

画像認識に必要とされる各種画像処理を高速に実行可能なマルチ SIMD アーキテクチャを提案し、シミュレーションによりその演算量の削減効果を示した。また、本アーキテクチャの実装に必要な共有メモリ回路の構成を示した。今後、より詳細な仕様を決定し画像認識システムのプロトタイプ試作を行う予定である。

## 文 献

- [1] S. Kyo, S. Okazaki and T. Arai: "An integrated memory array processor architecture for embedded image recognition systems", Proceedings of the 32nd International Symposium on Computer Architecture, pp. 134-145 (2005).
- [2] J. Tanabe, Y. Taniguchi, T. Miyamori, Y. Miyamoto, H. Takeda, M. Tarui, H. Nakayama, N. Takeda, K. Maeda and M. Matsui: "Visconti: Multi-VLIW image recognition processor based on configurable processor", Proceedings of the IEEE 2003 Custom Integrated Circuits Conference, pp. 185-188 (2003).
- [3] 石川, 小室: "デジタルビジョンチップとその応用", 電子情報通信学会論文誌 C, **J84-C**, 6, pp. 451-461 (2001).
- [4] J. Gealow and C. Sodini: "A pixel-parallel image processor using logic pitch-matched to dynamic memory", IEEE Journal of Solid-State Circuits, **34**, 6, pp. 831-839 (1999).
- [5] E. Gayles, T. Kellihg, R. Owens and M. Irwin: "The design of the MGAP-2: a micro-grained massively parallel array", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, **8**, 6, pp. 709-716 (2000).
- [6] 山本, 石井: "ポリゴンマッチングビジョンチップの設計", 電子情報通信学会論文誌 C, **J86-C**, 8, pp. 745-751 (2003).
- [7] T. Komuro, S. Kagami and M. Ishikawa: "A dynamically reconfigurable SIMD processor for a vision chip", IEEE Journal of Solid-State Circuits, **39**, 1, pp. 265-268 (2004).
- [8] 小室, 石川: "リアルタイム図形処理のための次元階層並列プロセッサ", ロボティクス・メカトロニクス講演会 2004 (ROBOMEC '04) 講演論文集, **2P2-L1-45**, (2004).