

マルチマスクオペレーションを効率よく実行する 画像処理用 LSI-ISP のアーキテクチャ†

福島 忠^{††} 小林 芳樹^{††} 平沢 宏太郎^{††}
坂東 忠秋^{†††} 柏岡 誠治^{††††} 加藤 猛^{†††††}

各種の濃淡画像を高速に処理できる画像処理用 LSI-ISP (Image Signal Processor) を開発した。ここでは、ISP の基本アーキテクチャに改善を加えることにより、テンプレート型の Prewitt オペレータのような、複数個のマスクデータを用いるマルチマスクオペレーションを、効率よく実行できることについて論ずる。アーキテクチャ上の改善は、プロセッサユニットとリンケージユニットにおけるマスク演算と、エバリュエーションユニットにおける比較演算とから検討した。その際、ISP における処理は、カーネル (局所演算領域) を切り出すごとに、そのカーネルに対するすべての演算を実行して、最終結果まで求めるものとし、PE (Processor Element) 増殖方式と PE 節約方式の二つのカーネル拡張方式に対処できるアーキテクチャを確立した。その結果、たとえば 4×4 画素のカーネルに対して 4 個のマスクデータを用いるマルチマスクオペレーションを、 256×256 画素の画像に施す場合、PE 増殖方式により ISP 4 個を用いて 43.7 ms で、PE 節約方式によれば ISP 1 個を用いて 174.8 ms で実行できることになった。

1. まえがき

濃淡画像処理におけるアルゴリズム研究は、古くからなされているが、その実用化は衛星画像や医療画像などの処理に限られている。そこで、一般産業への適用を目的として、各種の濃淡画像演算を高速に実行する多機能画像処理 LSI-ISP (Image Signal Processor) を開発した¹⁾。

ISP は、出力画像の一つの画素 (pixel) を算出するのに用いる入力画像の画素と、同数の PE (Processor Element) を用意して、局所的に並列演算する局所並列型に属し、次の特長を備えている。まず、空間積和演算に代表される局所近傍演算を、ビデオレートで高速処理できる。また、プログラマブル制御レジスタを内蔵し、2 値・濃淡・色彩画像の近傍画素を扱う基本演算を実行できる。さらに、局所演算領域 (kernel—以下カーネルと呼ぶ) を容易に拡張できる^{2), 3)}。

これらの特長を実現する ISP の基本アーキテクチャについては、空間積和演算を題材としてすでに報告

したが、マスクデータとしての荷重係数は 1 個しか用いられていない²⁾。しかし、2 個以上のマスクデータを用いるアルゴリズムも、これまでに数多く提案され、その有用性も実証されている。たとえば、8 個の差分型マスクを用いて、その最大出力値からエッジの強度や方向を求めるものとして、Prewitt⁴⁾、Kirsch⁵⁾、Robinson⁶⁾、などのテンプレート型オペレータがある⁷⁾。また、半導体チップの自動ワイヤボンディング装置では、複数のテンプレートを用いたパターンマッチングにより、 $\pm 15^\circ$ 程度の回転ずれを許容する位置検出を実現している⁸⁾。そこで、ISP においては、これまでに論じた基本アーキテクチャを基にして、複数個のマスクデータを用いる演算 (multi-mask operation—以下マルチマスクオペレーションと呼ぶ) を、効率よく実行できるよう最終的なアーキテクチャを決定した。

本論文では、ISP の基本アーキテクチャに改善を加え、カーネルの拡張性を損わずに、効率よくマルチマスクオペレーションを実行できるようにしたことについて論ずる。

2. ISP の基本アーキテクチャ

ISP の基本構成図と基本仕様を、それぞれ図 1 と表 1 に示す。図 1 に示すように、ISP は 6 個のユニットから構成されている。データユニットは、4 個の SR (Shift Register) などから成り、画像データの転送に寄与する。入出力バスは 8 ビット幅で、256 階調の濃淡データもしくは 8 個の 2 値データを扱える。

† Architecture of an Image Signal Processor for Multi-Mask Operations by TADASHI FUKUSHIMA, YOSHIKI KOBAYASHI, KOHTAROH HIRASAWA (The 10th Department, Hitachi Research Laboratory, Hitachi Ltd.), TADAOKI BANDO (The 8th Department, Hitachi Research Laboratory, Hitachi Ltd.), SEIJI KASHIOKA (The 6th Department, Central Research Laboratory, Hitachi Ltd.) and TAKESHI KATOH (The Control Computer Department Omika Works, Hitachi Ltd.).

†† (株)日立製作所日立研究所第 10 部

††† (株)日立製作所日立研究所第 8 部

†††† (株)日立製作所中央研究所第 6 部

††††† (株)日立製作所大みか工場計算制御設計部

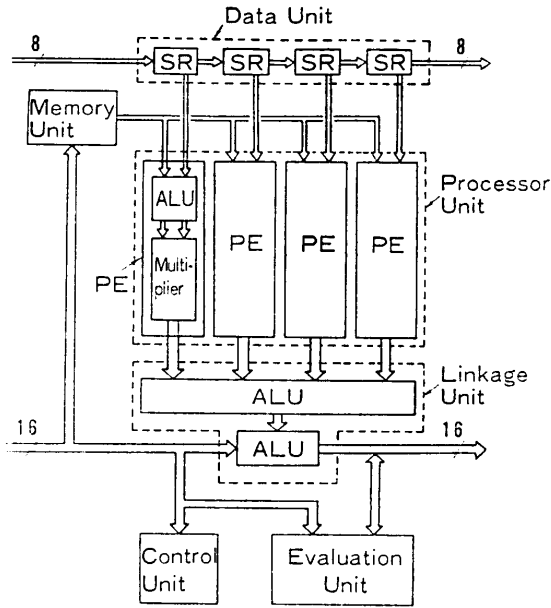


図1 画像処理用 LSI の基本構成
Fig. 1 ISP block diagram.

表1 画像処理用 LSI の基本仕様
Table 1 Specifications of the ISP.

テクノロジー	3 μm CMOS
トランジスタ数	~61,000
チップサイズ	7.72×8.64mm
電源電圧	5V
マシンサイクル	167ns
消費電力	~400mW
パッケージ	64 pin DIP

メモリユニットは、8bit×64word から成る RAM (Random Access Memory) で構成され、空間積和演算における荷重係数やパターンマッチングにおけるテンプレートなどのマスクデータを記憶する。画像演算時は、通常各 PE に1バイトずつデータを供給する。プロセッサユニットは4個の PE から構成され、各 PE は SIMD (Single Instruction Multi-Data stream) の形態で画像データを並列処理し、それぞれ16ビットの演算結果を、リンケージユニットへ出力する。リンケージユニットは、2個の ALU (Arithmetic Logic Unit) で構成され、PE 間と LSI 間の統合演算に寄与する。入出力データはすべて16ビットである。エバリュエーションユニットは、2個の比較器などから成り、リンケージユニットの出力データに対して、2値化やクラスタリングなどを行う。コントロールユニットは、プログラマブル制御レジスタなどから構成され、各ユニットを制御する。

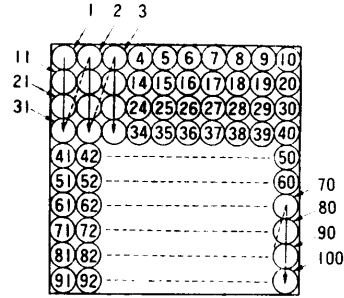


図2 小画像におけるスティック長4のスティック走査
Fig. 2 Stick scanning of length 4 for a small picture.

ISP の最大の特長は、二つの方式によりカーネルを拡張できることである。一つは、カーネルの大きさに合わせて使用する PE 数を増やす方式—PE 増殖方式—であり、もう一つは、PE を時分割に使用することにより、カーネルを形成する画素数より少ない PE で処理する方式—PE 節約方式—である。PE 増殖方式において処理される画像データは、通常のテレビ画像の走査方式であるラスタ走査 (raster scanning) により走査されればよい。一方、PE 節約方式においては、画像データは、筆者らが提案したスティック走査 (stick scanning) により走査されなければならない²⁾。

スティック走査とは、三つの走査方向から成る画像走査方式で、主走査方向は上から下、副走査方向は左から右、そして副々走査方向は上から下である。主走査方向において走査される画素の集合は、スティック (stick) と呼ばれ、一つのスティックに含まれる画素数は、スティック長 (stick length) と定義される。スティック長が1のスティック走査が、ラスタ走査となる。図2の小画像を例にとると、スティック長が4のスティック走査の場合、各画素は図2に示すように次の順に走査される。

- ①, ⑪, ⑳, ㉑, ②, ⑫, ㉒, ㉓, ③, ..., ⑩,
- ㉔, ㉕, ㉖, ⑪, ㉗, ㉘, ⑫, ㉙, ㉚, ⑬,
- ⑬, ..., ㉟, ㊱, ㊲, ㊳.

このスティック長が4のスティック走査を用いると、PE 節約方式により、1個の ISP でカーネルが4×4画素の空間積和演算を実行できる。このときのISP のアーキテクチャを図3に示す。

図3では、各 SR の遅延段数は、スティック長に等しい4段となっている。つまり、この3個の SR に3本のスティック (合計12画素) が一時記憶されることになる。また、RAM には、あらかじめ PE 1個当たり4個の荷重係数を書き込んでおく。ここで、次の

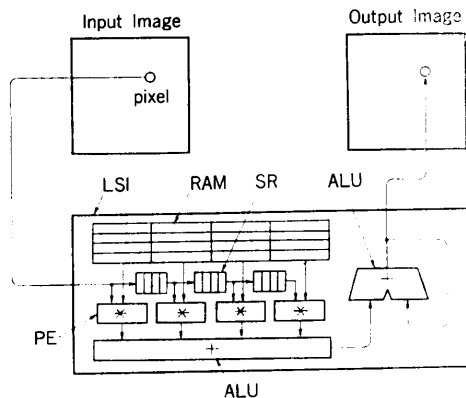


図 3 PE 節約方式における 4×4 空間積和演算実行のための構成

Fig. 3 Architecture for 4×4 spatial convolution with the PE-save system construction.

スティック (4 画素) を 4 回に分けて左端から入力すると、SR からの出力も含めて、4×4 のカーネルを形成する 16 個の画素が、4 個の PE に供給されることになる。16 個の画素は、RAM から読み出される荷重係数と、4 回に分けて掛け合わされた後、二つの ALU で加算されて出力画素データとなる。後段の ALU は、アキュミュレータとして動作することになる。

3. マルチマスクオペレーション

ある原画像からカーネルを形成する画素を切り出して、局所近傍演算を施す場合、通常、マスクデータとのマスク演算となる。たとえば空間積和演算では、マスクデータとして荷重係数が用意され、画素データは、荷重係数と掛け合わされた後加算される。このとき、一つの画像処理で用いられるマスクデータが、平滑化オペレータやラプラシアンオペレータなどの単一のもの、Prewitt オペレータのテンプレート型のような複数ものものと分けられる。ここでは、後者を前者と区別して、マルチマスクオペレーションと呼んでいる。代表的なマルチマスクオペレーションを表 2 に示す。

表 2 のなかで、Prewitt, Kirsch, Robinson のテンプレート型オペレータはよく知られている。これらは、3×3 のカーネルに 8 個の差分型マスクを掛け合わせ、その最大値をエッジの強度として、また、そのときのマスクが M_i ならば、 i をエッジの方向として出力するものである。

このようなマルチマスクオペレーションの処理方式

は、次の二つの観点から分類できる。第一は、一つのマスクデータに関する演算を全画面にわたって実行することを、マスクデータの数と等しい回数繰り返すのか、それとも、一つのカーネルを切り出すごとに、用意したすべてのマスクデータとの演算を実行することを、全画面にわたって繰り返すのか、という点である。第二は、個々のカーネルに対して、用意したすべてのマスクデータとの演算を終えて、マスクデータと同数の演算結果を求めた後、その演算結果間の演算を一度に実行するのか、もしくは、一つのカーネルを切り出し、一つのマスクデータとの演算を終了するごとに、マスク演算結果間の演算も実行してゆくのか、という点である。

上記二つの観点の組合せから、マルチマスクオペレーションの処理形態は 4 通り考えられるが、ISP においては、カーネルを切り出すごとに、すべてのマスク演算を行うと同時に、それらの演算結果の処理まで行って、一つのカーネルに対する最終結果まで求めることにした。なぜなら、より効率的なシステムを構築するためには、画像メモリの参照を最小限にすることが望ましいからである。つまり上記の方法によると、入力画像の読出しと最終結果の書き込みの、2 回のメモリ参照でマルチマスクオペレーションを処理でき、メモリ参照によるオーバーヘッドを最小限に抑えることができる。

なお、表 2 からわかるように、マルチマスクオペレーションにおけるマスク演算結果間の処理の多くは、比較演算によりなされる。そこで、エバリュエーションユニットに比較器を設け、各種の 2 値化処理と合わせて、マスク演算結果間の処理を行わせることにした。比較演算で算出できないマスク演算結果間の処理については、チップ面積などの理由により、ISP で実現するのは断念した。

4. アーキテクチャの検討

4.1 アーキテクチャ上の課題

3 章に述べた方針に基づいて、ISP の基本アーキテクチャに対する改善を検討してゆくうえで、次の四つの課題があると考えられる。

- (1) 画像データの PE への供給方式。
- (2) マスクデータの PE への供給方式。
- (3) マスク演算結果間の比較演算方式。
- (4) 比較演算結果の出力方式。

(1) と (2) の課題は、PE における演算の順序と、

表 2 代表的なマルチマスクオペレーション
Table 2 Representative multi-mask operations.

入力画像	名 称	内 容	マ ス ク	デ	ー	タ				
2 値	複合パターンマッチング	複数個の標準パターンと照合して、最もよく一致した標準パターンと、そのときの一致度を求める。	M0	M1	M2	M3	M4	M5	M6	M7
	Prewitt オペレータ (テンプレート型)		$\begin{matrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{matrix}$	$\begin{matrix} 1 & 1 & 1 \\ 1 & -2 & -1 \\ 1 & 1 & -1 \end{matrix}$	$\begin{matrix} 1 & 1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & -1 \end{matrix}$	$\begin{matrix} 1 & -1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & 1 \end{matrix}$	$\begin{matrix} -1 & -1 & -1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{matrix}$	$\begin{matrix} -1 & -1 & 1 \\ -1 & -2 & 1 \\ 1 & 1 & 1 \end{matrix}$	$\begin{matrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{matrix}$	
	Kirsch オペレータ	M0~M7 のマスクを 3x3 のカーネルに掛け合わせ、その最大値をエッジ強度とし、そのときのマスクが Mi ならば、エッジ方向を i とする。	$\begin{matrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & 3 \end{matrix}$	$\begin{matrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{matrix}$	$\begin{matrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{matrix}$	$\begin{matrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{matrix}$	$\begin{matrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{matrix}$	$\begin{matrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{matrix}$	$\begin{matrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{matrix}$	$\begin{matrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{matrix}$
濃 淡	Robinson オペレータ		$\begin{matrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{matrix}$	$\begin{matrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{matrix}$	$\begin{matrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{matrix}$	$\begin{matrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{matrix}$	$\begin{matrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{matrix}$	$\begin{matrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{matrix}$	$\begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix}$	$\begin{matrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{matrix}$
	Prewitt オペレータ (ディファレンシャル型)		$\begin{matrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{matrix}$	$\begin{matrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{matrix}$						
	Sobel オペレータ		$\begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix}$	$\begin{matrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{matrix}$						
色 彩	色彩距離による分類	複数個の標準色と照合して、色彩距離がある許容範囲のとき、その標準色として分類する。								

(エッジ強度) = $\sqrt{SX^2 + SY^2}$ or $|SX| + |SY|$

(エッジ方向) = $\tan^{-1}(SY/SX)$

$D_i = \sqrt{(R - \alpha_i)^2 + (B - \beta_i)^2 + (G - \tau_i)^2}$ or $|R - \alpha_i| + |B - \beta_i| + |G - \tau_i|$

D_i : クラス i の標準色からの色彩距離

R, B, G : 入力画像の光の 3 原色の各濃度値

$\alpha_i, \beta_i, \tau_i$: クラス i の標準色の光の 3 原色の各濃度値

リンケージユニットにおける演算の制御に係わる問題である。一方、(3)と(4)の課題は、エバリュエーションユニットにおける演算に関する問題であり、(1)と(2)から切り離して考えることができる。なぜなら、3章に述べた方針に基づくと、カーネルごとの各マスクデータに対する演算結果は、リンケージユニットからエバリュエーションユニットへ順次供給され、(1)と(2)の方式が異なっても、転送のタイミングが多少違って来るだけだからである。そこで、(1)、(2)の検討と、(3)、(4)の検討を、分けて考察する。

ここで考慮しなければならないのは、二つのカーネル拡張方式への対処である。なぜなら、PE 増殖方式と PE 節約方式では、画像走査方式が異なるからである。PE 増殖方式ではラスタ走査が用いられ、PE 節約方式ではスティック走査が用いられる。

しかし、ラスタ走査はスティック走査の特殊なものの一スティック長が1のスティック走査であるから、スティック走査により走査された画像データを処理できれば、ラスタ走査により走査された画像データも処理できることになる。つまり、PE 節約方式に対処できれば、PE 増殖方式にも対処できるといえる。そこで、PE 節約方式によるカーネル拡張に対して、上記の課題を検討することにする。

4.2 データ供給に関する検討

ここでは、画像データとマスクデータの PE への供給方式について検討するが、討論を簡潔にするため、スティック長が2のスティック走査を用いて、2個のマスクデータを扱うマルチマスクオペレーションを実行する場合について考察する。

上記の処理を実行する際、一つの PE とそれに関連する部分だけを抜粋した模式図を、図4に示す。同図において、P1とP2は画素データを、A1とA2は1個目のマスクデータを、B1とB2は2個目のマスクデータを表している。なお図4においては、空間積和演算を想定して演算機能を選択しているため、リンケージユニットの ALU からは、演算結果として次の二つの値が出力される。

$$R1 = A1 * P1 + A2 * P2.$$

$$R2 = B1 * P1 + B2 * P2.$$

PE への画像データとマスクデータの供給方式は、PE における演算の実行順序に左右される。上記の演算では、一つの PE で4回の乗算が実行されなければならないが、次の二つの順序が考えられる。

- (1) $A1 * P1, A2 * P2, B1 * P1, B2 * P2.$

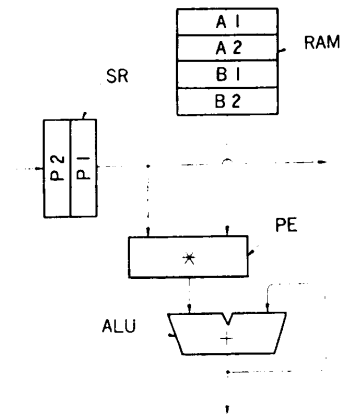


図4 スティック長2、マスクデータ数2のマルチマスクオペレーションの模式図

Fig. 4 Architecture model for multi-mask operations of two masks with the stick scanning of length two.

- (2) $A1 * P1, B1 * P1, A2 * P2, B2 * P2.$

つまり、(1)は、マスクデータごとに演算を進める方式であり、(2)は、画像データごとに演算していく方式である。

(1)の方式では、マスクデータ単位の演算を、マスク数だけ繰り返すことになり、演算の制御は容易であるが、同じ画素データを繰り返し入力する必要がある。このとき、SRの遅延段数は、(スティック長) × (マスク数) となる。図4の例では、SR 当り4段にすればよいが、データユニットには4個のSRがあることや、スティック長やマスク数の増加を考慮すると、SRのゲート数は膨大なものになる。たとえば、表2にある Prewitt オペレータのテンプレート型を処理する場合、各SRの遅延段数は24段になる。

(2)の方式では、個々の画素データは、マスク数に等しい回数だけ続けて使用されるため、何度も入力する必要はない。マスク数が増加すれば、それに応じて、SRの転送レートを低下すればよいので、SRの遅延段数は、つねにスティック長と同数でよい。しかし、それぞれのマスクデータに対する演算結果を算出するためには、リンケージユニットに、マスク数と同数の一時記憶バッファを用意して、複雑なデータ制御を施さなければならない。

(1)、(2)の長所と短所を比較検討した結果、ISPにおいては、下記の理由により(1)の方式を採用することにした。まず、制御が容易であること。次に、基本アーキテクチャをほとんど変更する必要がないこと。さらに、SRの段数増加は、図5に示す方式によ

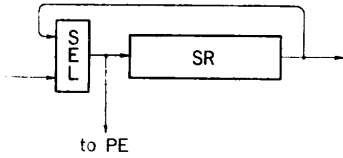


図5 シフトレジスタにおけるフィードバック機構
Fig. 5 Feedback mechanism of shift registers.

り解決され、若干のゲート数増で対応できること、からである。

図5において、PEへ転送される画素データは、同時にSRの初段にフィードバックされ、マスク数に応じて再度PEへ転送されることになる。一方、マスクデータは、RAMが外部端子から直接アドレスされるため、(1)の方式を採用することによる不都合は何もない。

なお、それぞれの画素データには、2ビットのシーケンスコード(sequence code)を付加して、その画素データがスティック内のどこに位置するかという情報をもたせることにした。このシーケンスコードが、SRのフィードバック機構や、リンケージユニットにおける統合演算を制御することになる。

これらの結果、それぞれのマスクデータに対する演算結果は、スティック長に等しいマシンサイクルごとに、リンケージユニットから出力されることになる。

4.3 比較演算に関する検討

3章の方針に基づくと、エバリュエーションユニットは、比較器を中心とした構成になる。そこで、エバリュエーションユニットの構成は、比較器を用いる閾値処理(thresholding)を基本として、マルチマスクオペレーションにおけるマスク演算結果間の、比較演算を実現することから検討した。

閾値処理には、閾値が固定されたままの固定閾値処理と、処理の進行に伴って閾値の変化する浮動閾値処理とがあるが、浮動閾値処理は、カーネル内の演算と固定閾値処理などに変換できるものもあるので、ISPにおいては、固定閾値処理についてのみ考察することにした。

固定閾値処理においては、参照データは固定の閾値により2値化されるが、より汎用性を高めるために、二つの閾値 $T_H, T_L (T_H > T_L)$ を用いた4種類の2値化処理を実現することにした。参照データを f 、出力結果を g とすると、4種類の2値化処理は次のように表される。

Ⓐ $g=1$ if $f > T_H$, otherwise $g=0$.

Ⓑ $g=1$ if $f \leq T_L$, otherwise $g=0$.

Ⓒ $g=1$ if $f > T_H$ or $f \leq T_L$, otherwise $g=0$.

Ⓓ $g=1$ if $T_L < f \leq T_H$, otherwise $g=0$.

これらを実行するためには、2個の比較器と2個の閾値レジスタが必要であるとともに、任意の閾値を用いるように、演算の実行に先立って閾値レジスタの内容が自由に書き替えられることが重要である。さらに、比較結果により(たとえば $g=1$ のとき)、参照データで閾値レジスタの内容を書き替えられるならば、参照データの最大値や最小値を求めることができる。また、2値化情報と合わせることにより、最大値・最小値の座標値を求めることも可能である。

一方、マルチマスクオペレーションにおけるマスク演算結果間の比較演算を実現するには、次の二つの機能が要求される。

(1) マスク演算結果のうちの最大値もしくは最小値を、カーネルごとに抽出する。

(2) (1)で求めた最大値もしくは最小値を生み出したマスクデータ番号を作り出す。

これらのことを踏まえて、エバリュエーションユニットの構成を、図6に示すように決定した。

図6の構成において、リンケージユニットから与えられる参照データは、2個の比較器COMPに同時に入力される。それぞれの比較器は、通常、閾値レジスタ(thresholding register)の内容と参照データとを比較し、大小の判定結果を出力する。判定結果は、最終的には1ビットの2値データとして、LSI外部に出力される。また、比較器からは、比較した二つのデータの大きいほう(もしくは小さいほう)の値を閾値レ

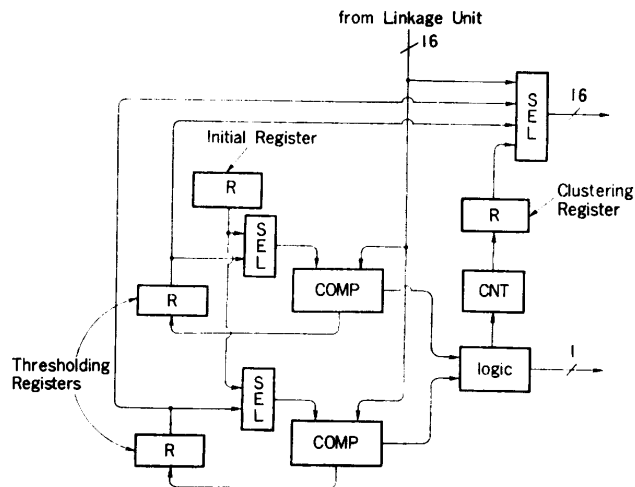


図6 エバリュエーションユニットの構成
Fig. 6 Architecture of the evaluation unit.

レジスタに戻す。閾値レジスタは書替え可能になっており、機能に応じて比較器の出力データで更新される。2個の比較器の一方は大きい値を、他方は小さい値を抽出するよう機能が固定されているので、閾値レジスタの書替えを停止することにより、前述した4通りの固定閾値処理を実行できる。また、閾値レジスタの書替えを許すことにより、参照データの最大値や最小値を求めることができる。

しかし、これだけでは、(1)のカーネルごとにマスク演算結果間の最大値・最小値を求めることはできない。なぜなら、カーネルが移動するごとに、閾値レジスタを初期化しなければならないからである。一方、一つのカーネルにおけるマスク演算結果間の処理が終了すると、その結果を LSI 外部へ出力しなければならないので、それが終了するまで閾値レジスタを初期化できない。つまり、閾値レジスタの内容の出力と初期化のために、一つのカーネルについての処理を終了した後、次のカーネルについての処理の開始を、遅らせる必要が生ずる。この問題を解決するために、図6に示す初期値レジスタ (initial register) を設けた。

新たなカーネルの最初の参照データが比較器に入力される時、閾値レジスタに代わって、初期値レジスタの内容が比較器に与えられる。このとき、どちらか一方の閾値レジスタの内容が、LSI 外部に出力される。閾値レジスタの内容が出力された後、初期値レジスタと参照データとの比較結果が、閾値レジスタに書き込まれる。このように、初期値レジスタを設けて閾値レジスタの初期化を省略することにより、新たなカーネルについての処理の開始の遅延を防ぐことが可能となった。なお、図6では繁雑を避けるため記入していないが、二つの閾値レジスタと初期値レジスタは、リンケージユニットの入力バスを介して、任意の値に設定することができる。

マルチマスクオペレーションのマスク演算結果間の処理を実現するには、前述の(2)の機能、すなわち、求めた最大値・最小値を生み出したマスクデータ番号を生成できなければならない。これを生成するのが、図6に示すバイナリカウンタ CNT である。バイナリカウンタは、カーネルが移動するたび初期化され、マスク演算結果が比較器に与えられるごとにカウントアップする。バイナリカウンタの内容は、比較器の判定結果により、適宜クラスタリングレジスタ (clustering register) に書き込まれる。

これらの結果、一つのカーネルにおけるマスク演算

処理が終了したとき、マスク演算結果の最大値もしくは最小値は閾値レジスタに、その値に関するマスク番号はクラスタリングレジスタに残される。これらの値は、リンケージユニットの出力バスを介して、時分割で LSI 外部に出力される。なお、図6において、閾値レジスタと初期値レジスタの出力選択や、バイナリカウンタの初期化のタイミングなどの制御には、データユニットやリンケージユニットにおける制御と同様に、各画素データに付加されて入力される2ビットのシーケンスコードを用いている。

5. アーキテクチャ改善の効果

4章で述べたアーキテクチャの改善を実施することにより、次の二つの効果がある。まず、周辺回路の追加なしで、ISP だけで、表2に示すマルチマスクオペレーションを実行できる(ただし、平方根、三角関数などは除く)。さらに、マスク数の増加にかかわらず、画像メモリのアクセスは、読出しと書込みがそれぞれ1回でよい。これらにより、マルチマスクオペレーションを実行できる画像処理システムを、少数の ISP を用いて、非常にコンパクトに構築できる。

ISP では、メモリユニットに、一つの PE 当り 16 バイトのマスクデータを格納できる。このため、PE 増殖方式によりカーネルを拡張した場合、16個のマスクデータまで処理できる。PE 節約方式による場合、 $[16/(\text{スティック長})]$ 個のマスクデータまで扱える。たとえば、スティック長が4のとき、扱えるマスクデータ数は4である。

最後に、カーネルが 4×4 画素で、4個のマスクデータを用いるマルチマスクオペレーションを、 256×256 画素の画像に施す場合について述べる。PE 増殖方式の場合は、ISP を4個必要とするが、1画像に要する処理時間は約 43.7 ms である。一方、PE 節約方式では、ISP は1個でよいが、必要な処理時間は約 174.8 ms である。

6. むすび

画像処理用 LSI-ISP (Image Signal Processor) は、Prewitt オペレータのテンプレート型のように、複数個のマスクデータを用いるマルチマスクオペレーションを、効率よく実行できる。

ISP においては、カーネルを切り出すごとにマスク演算結果間の処理まで行うことにし、そのアーキテクチャを、画像データとマスクデータの PE への供給方

式と、マスク演算結果間の処理方式に分けて検討した。前者については、画像データを繰り返し PE へ供給するため、データユニットの SR にフィードバック機構を設けるとともに、マスクデータを格納するため、メモリユニットの記憶容量を増やした。また後者については、エバリュエーションユニットに、初期値レジスタとバイナリカウンタを設け、閾値レジスタを書き換え可能とした。これらにより、効率のよい比較演算と、その結果の出力が可能となった。

今後の課題は、ISP の特長を十二分に活かした、効果的な画像処理システムの構築法の開発である。

謝辞 本研究をご支援下さった(株)日立製作所大みか工場桑原洋副工場長、有益なご討論をいただいた同社中央研究所江尻正員主管研究員、同社生産技術研究所秦清治主任研究員、および LSI 開発にご協力いただいた諸氏に深く感謝する。

参 考 文 献

- 1) Fukushima, T. et al.: An Image Signal Processor, *IEEE International Solid State Circuits Conference Digest of Technical Papers*, Vol. 26, pp. 258-259 (1983).
- 2) 福島, 小林他: 画像処理用 LSI-Image Signal Processor のアーキテクチャ, *信学論(C)*, Vol. J 66-C, No. 12, pp. 959-966 (1983).
- 3) 福島, 小林他: 多機能性を実現する画像処理用 LSI-ISP のアーキテクチャ, *情報処理学会論文誌*, Vol. 25, No. 5, pp. 728-735 (1984).
- 4) Prewitt, J. M. S.: Object Enhancement and Extraction, in: *Picture Processing and Psychopictorics*, pp. 75-149, Academic Press, New York (1970).
- 5) Kirsch, R.: Computer Determination of the Constituent Structure of Biological Images, *Comput. Biomed. Res.*, Vol. 4, pp. 315-328 (1971).
- 6) Robinson, G. S.: Edge Detection by Compass Gradient MAsks, *CGIP*, Vol. 6, pp. 492-501 (1977).
- 7) 坂根, 田村: SPIDER 開発を通して見たデジタル画像処理アルゴリズムの現状(3), *情報処理学会コンピュータビジョン研究会資料*, 4-4 (1980).
- 8) 柏岡, 江尻, 坂本: 時分割パターン認識技術による群制御トランジスタ組立システム, *電学論C*, Vol. 96, No. 1, pp. 9-16 (1976).

(昭和 59 年 3 月 21 日受付)

(昭和 59 年 9 月 20 日採録)