画像論理代数(ILA)とその応用†

福井将樹†† 北山研一††

光の並列性を生かしたSIMD型の並列コンピュータにおいては，その並列演算能力を有効に利用するための演算アルゴリズムが重要である．本論文では，光の並列性を利用した演算アルゴリズムを実行するための体系である画像論理代数(ILA)を提案し，その定義について述べる．さらに，応用のための基本的演算となる整数加算および乗算のILAによる実現方法を示す．整数加算については二種類のアルゴリズムを示し，効率について比較した．整数乗算については，ブースのアルゴリズムを用いることにより従来の光並列乗算よりも効率的なアルゴリズムが得られた．最後にILAに基づいた光コンピュータのアーキテクチャを示した．

1. はじめに

光は自由空間において，その2次元的な波面形状を保存した状態で伝播する．この光の並列伝播性をデータチャンネルとして利用すれば，多くのデータ(画像)に対して同一の処理をおこなうSIMD型の並列処理を実現できる可能性がある．そのため従来より，様々なSIMD型の並列光コンピュータアーキテクチャが提案されている[1),2)]．これらのアーキテクチャ上で実行される画像処理や数値演算のアルゴリズムは，それぞれ符号置換論理[1),3)]，光アレイロジック[4)]，二値画像代数*(BIA)[5)]等の光演算法により記述される．これらの光演算法における基本的な演算の一つは特定の近傍画素状態（演算用パターン）を抽出する演算すなわち画像と演算用パターンの相関演算である．この演算用パターンの記述法は，アルゴリズムの効率的な開発手法を確立する上で重要である．

本論文では，光によるSIMD型の並列演算に適した演算体系として画像論理代数(Image Logic Algebra, ILA)を提案する．次に，様々な応用のための基本的演算として，加算および乗算の具体的なアルゴリズムをILAで記述できることを示す，最後に，ILAに適した光コンピュータのアーキテクチャを示す．ILAの特徴は，新しい概念としてNCPと呼ぶ演算用パターンとその積，和，否定という新しい概念を導入することにより，従来十分な検討がなされていなかった演算用パターンの記述法を確立したことである．これによって，符号置換論理や光アレイロジック，二値画像代数等の並列演算アルゴリズムを包括的に分かりやすい，一般的な形式で記述することが可能となる．

† Image Logic Algebra (ILA) and Its Applications by MASAKI FUKUI and KEN-ICHI KITAYAMA (NTT Transmission Systems Laboratories, Lightwave Communications Laboratories).

†† NTT伝送システム研究所光通信研究部

2. 画像論理代数(ILA)

ILAは，**表1**に示すように二値画像および新たに導入した近傍画素の状態を表す演算用パターンの概念であるNCP (Neighborhood configuration pattern)に対する12種類の演算および変換から構成される．以下において，被処理画像の定義および画像に対する演算（論理積，論理和，否定）および変換（シフト，Image Casting, Multiple Imaging, テスト）について述べ，さらにNCPとNCPに対する演算（論理積，論理和，否定）および，NCPを用いた演算extended erosion, dilationについて述べる．

2.1 画像に対する演算

ILAが対象とする画像は，各画素が論理値0または1をもつ二値画像である．$A[M;N]$は横がM画素，縦がN画素の二値画像をあらわす．$A[M;N]$における座標(i,j)の画素値を$A[M;N]_{ij}$($1\leq i\leq M$, $1\leq j\leq N$)とあらわす．座標は左上を原点とし，右および下が正の方向である．一方，二値画像はシンボリックにも表現される．例として**図1**の画像は次式で表現される．

$$A[3;3]=\begin{vmatrix}1&0&0\\0&1&1\\1&0&0\end{vmatrix},\quad B[2;3]=\begin{vmatrix}1&0\\0&0\\1&1\end{vmatrix}. \qquad (1)$$

画像$A[M;N]$, $B[M;N]$の論理積AND, 論理和ORはそれぞれ対応する画素間の論理積，論理和として定義され，$A[M;N]\cdot B[M;N]$, $A[M;N]+B[M;N]$と表記する．画像$A[M;N]$の否定は各画素の否定として定義され$\overline{A[M;N]}$と表記する．

2.2 画像に対する四つの変換

2.2.1 シフト

画像$A[M;N]$から$B[M;N]$へのシフト変換は$B[M;N]=\delta_{kl}A[M;N]$と表記し，次式で定義され

表 1 ILA の演算と変換
Table 1 Operations and transformations of ILA.

演算	画像間	論理積，論理和
	画 像	否 定
	NCP 間	論理積，論理和
	NCP	否 定
変換	画 像	シフト，テスト，*IC*, *MI*
	画像-NCP	extended erosion, dilation

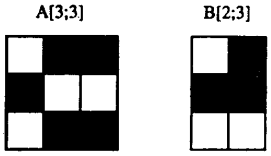


図 1 二値画像の例
Fig. 1 Example of binary images.

る．

$$B[M;N]_{ij}=A[M;N]_{i+k,j+l}. \quad (2)$$

2.2.2 Image Casting

画像 $A[M;N]$ から $B[R;S]$ への Image Casting 変換は $B[R;S]=IC(A[M;N])$ と表記し，次式で定義される．

(i) $R=kM$, $S=lN$ $(k,l=2,3,\cdots)$ の時

$$B[R;S]_{mn}=\begin{cases}A[M;N]_{ij} & m=ki, n=lj\\ 0 & \text{otherwise.}\end{cases} \quad (3)$$

(ii) $M=kR$, $N=lS$ $(k,l=2,3,\cdots)$ の時

$$B[R;S]_{ij}=A[M;N]_{ki,lj}. \quad (4)$$

図 2 に次式の Image Casting 変換の例を示す．

$$A[6;6]=IC(A[3;3])$$
$$B[2;2]=IC(B[4;4]). \quad (5)$$

2.2.3 Multiple Imaging

画像 $A[M;N]$ から $B[R;S]$ (ただし $R=kM$, $S=lN$ $(k,l=1,2,3,\cdots)$ への Multiple Imaging 変換は $B[R;S]=MI(A[M;N])$ と表記し，次式で定義される．

$$B[R;S]_{i+mM,j+nN}=A[M;N]_{ij}$$
$$m=1,2,\cdots,k, \text{ and } n=1,2,\cdots,l. \quad (6)$$

図 3 に次式の Multiple Imaging 変換の例を示す．

$$A[8;8]=MI\left(\left|\begin{matrix}0&1&0&0\\0&1&0&0\\0&1&1&1\\0&0&0&0\end{matrix}\right|\right). \quad (7)$$

2.2.4 テ ス ト

画像 $A[M;N]$ から $B[1;1]$ へのテスト変換は

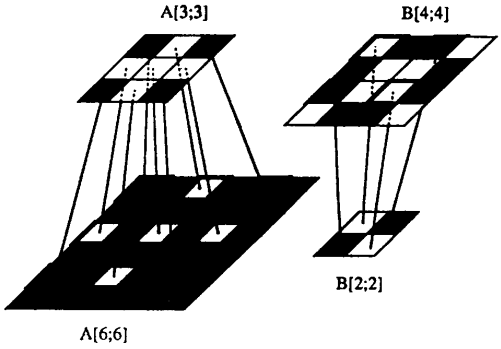


図 2 Image Casting 変換の例
Fig. 2 Example of Image Casting transformation.

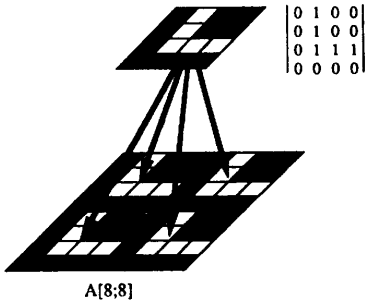


図 3 Multiple Imaging 変換の例
Fig. 3 Example of Multiple Imaging transformation.

$B[1;1]=TEST(A[M;N])$ と表記し，次式で定義される．

$$B[1;1]_{1,1}=\prod_{i=1}^{M}\prod_{j=1}^{N}\overline{A[M;N]_{ij}}. \quad (8)$$

2.3 NCP とその演算

ILA における extended erosion は数理形態学における erosion を拡張したものであり，NCP が structuring element に対応する．

NCP はある画素を中心とする近傍画素の状態を表すパターンである．図 4 に NCP の例を示す．0, 1 はそれぞれ画素値 0, 1 を示し，＊は画素の状態が指定されていないことを示す．また，中心画素は下線で示す．図 4 (a) は中心画素が 1，左上および右上が 0 の近傍画素状態を表す NCP であり，図 4 (b) は中心画素が 1，左および右が 0 の近傍画素状態を表す NCP である．また，図 4 (c) は中心および左の画素が 1，右の画素が 0 の近傍画素状態を表す NCP である．

次に NCP の論理積，論理和，否定について述べる．二つの NCP の論理積 ^ は，その二つの NCP が示す近傍画素状態が同時に存在する NCP を生成する演算である．例えば，図 4 (a) と (b) の NCP の論理積を次式に示す．

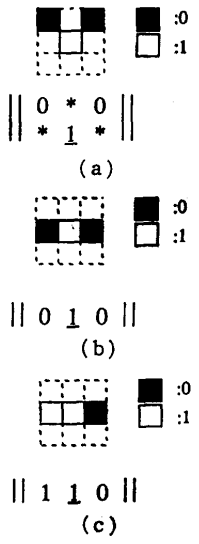


図 4 NCP の例
Fig. 4 Examples of NCP.

$$\left\|\begin{matrix}0 & * & 0\\ 0 & \underline{1} & 0\end{matrix}\right\| = \left\|\begin{matrix}0 & * & 0\\ * & \underline{1} & *\end{matrix}\right\| \wedge \|0\ \underline{1}\ 0\|. \tag{9}$$

二つの NCP の論理和 $\vee$ は，その二つの NCP が示す近傍画素状態のいずれか一方が存在する NCP を生成する演算である．例えば，図 4 (b)と(c)の NCP の論理和を次式に示す．

$$\|\underline{1}\ 0\| = \|0\ \underline{1}\ 0\| \vee \|1\ \underline{1}\ 0\|. \tag{10}$$

ここで，NCP を論理和で結合したものも NCP であるとする．(10)式の例では，二つの NCP は一つの画素状態すなわち中心から一つ左にある画素状態のみ異なっていたため，一つの NCP に縮退したが，一般的には一つの NCP になるとは限らない．

NCP の否定は，その NCP が示す近傍画素のどの一つの画素状態も存在しない NCP を生成する演算である．例えば，図 4 (c)の NCP の否定を次式に示す．

$$\|0\ \underline{*}\| \vee \|0\| \vee \|\underline{*}\ 1\| = \overline{\|1\ 1\ 0\|}. \tag{11}$$

次式で示す NCP $\boldsymbol{P}$ について考える．

$$\boldsymbol{P} = \left\|\begin{matrix}\ddots & \vdots & \vdots & \vdots & \iddots\\ \cdots & r_{-1,1} & r_{0,1} & r_{1,1} & \cdots\\ \cdots & r_{-1,0} & \underline{r_{0,0}} & r_{1,0} & \cdots\\ \cdots & r_{-1,-1} & r_{0,-1} & r_{1,-1} & \cdots\\ \iddots & \vdots & \vdots & \vdots & \ddots\end{matrix}\right\|. \tag{12}$$

ここで，$r_{k,l}$ は 1, 0, * のいずれかの値をとるものとする．NCP の論理積の定義から(12)式は次式のように書ける．

表 2 r_{klm} の定義
Table 2 The relationship of r_{klm} and x_{klm}.

r_{klm}	$x^{(0)}_{klm}$	$x^{(1)}_{klm}$
1	0	1
0	1	0
*	1	1

$$\boldsymbol{P} = \bigwedge_{k,l} \|r_{kl}\|_{(k,l)}. \tag{13}$$

さらに，NCP を論理和で結合したものも NCP であると定義したため，NCP の一般形は次式で表記できる．

$$\boldsymbol{S} = \bigvee_{m} \bigwedge_{k,l} \|r_{klm}\|_{(k,l)}. \tag{14}$$

2.4 Extended erosion

NCP $\boldsymbol{S}$ により画像 $A[M; N]$ を extended erosion した画像を $B[M; N]$ とし，次式で表記する．

$$B[M; N] = A[M; N] \otimes \boldsymbol{S}. \tag{15}$$

ここで，

$$B[M; N]_{ij} = \sum_{m} \prod_{k,l} x^{(0)}_{klm} A[M; N]_{i+k,j+l} + x^{(1)}_{klm} \overline{A[M; N]_{i+k,j+l}} \tag{16}$$

である．ただし，$x^{(0)}_{klm}$，$x^{(1)}_{klm}$ と r_{klm} は表 2 の関係にある．NCP 間の論理和 $\vee$，論理積 $\wedge$ と extended erosion との関係を次に示す．

$$A[M; N] \otimes (S_1 \vee S_2) = A[M; N] \otimes S_1 + A[M; N] \otimes S_2 \tag{17}$$

$$A[M; N] \otimes (S_1 \wedge S_2) = (A[M; N] \otimes S_1) \cdot (A[M; N] \otimes S_2) \tag{18}$$

2.5 Dilation

画像 $A[M; N]$ に対する NCP $\boldsymbol{P}$ による dilation を次式で定義する．

$$A[M; N] \oplus \boldsymbol{P} \equiv \overline{A[M; N] \otimes \left(\bigwedge_{k,l} \overline{\|r_{kl}\|_{(-k,-l)}}\right)}. \tag{19}$$

ここで，NCP $\boldsymbol{P}$ は(13)式で与えられたものである．ここで，(19)式で定義される dilation は，NCP $\boldsymbol{P}$ を structuring element とすれば数理形態学における dilation と同一である．証明は省略するが，(19)式に(16)式を適用し計算をおこなうことにより証明できる．

3. 加算および乗算の実現

多値画像処理では，二次元に配列した数値に対する処理が基本となる．そこで，ILA による二次元データに対する数値計算の方法を示す．具体的には，二の補数表現された整数に対する加算，乗算の ILA による実行方法を示す．

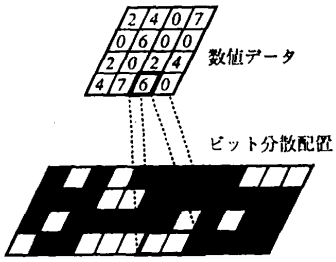


図 5 ビット分散配置
Fig. 5 Binary row-coded numbers.

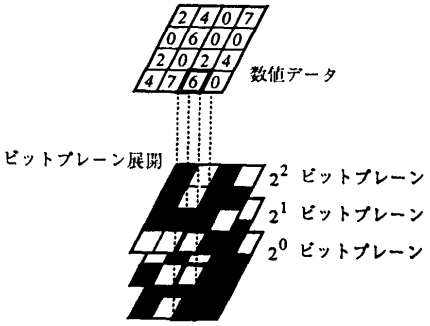


図 6 ビットプレーン配置
Fig. 6 Binary stack-coded numbers.

光学的な方法で二値画像間の論理演算を実行することを考えると，二値画像を空間符号化しマスク処理により論理演算をおこなう方法（スペースバリアント論理演算法[6]）が処理の自由度の点に関して有利であるが，応用に際してのマスクパターンの決定方法が未検討であった．そこで，マスクパターンを演算データによって動的に決定する方法を新たに提案し，その手法を用いたブースのアルゴリズムによる整数乗算法をILA により記述する．

ILA では数値データも二次元二値パターンとして表現しなければならない．数値を二値パターンとして表現する方法として大きく分けて，ビット分散配置による方法[5]（図 5）とビットプレーン配置による方法[5]（図 6）が知られている．ビット分散配置による方法では，数値データの各桁ビットを同時に処理できる可能性はあるが同時に処理できるデータの数が数値のビット長に反比例して少なくなる．一方，ビットプレーン配置では，同時に処理できるデータの数は数値のビット長に依存しないが，一つのデータの各ビットを同時に処理できない．以下の ILA による数値演算法では，同時に処理できるデータの数が多いビットプレーン配置を用いる．

3.1 空間符号化

空間符号化は二枚の二値入力画像の対応画素を図 7 の符号化ルールにしたがって空間符号化する処理である．ILA による空間符号化の処理手順を次に示す．ここで，$A[N;N]$，$B[N;N]$ は二枚の入力画像であり，$C[2N;2N]$ は符号化画像とする．

$$A[2N;2N]=IC(A[N;N])$$

$$B[2N;2N]=IC(B[N;N])$$

$$A[2N;2N]=A[2N;2N]\oplus\left\|\begin{matrix}* & 1\\ * & 1\end{matrix}\right\|+\overline{A[2N;2N]}\oplus\left\|\begin{matrix}1\\ 1\end{matrix}\right\|$$

$$B[2N;2N]=B[2N;2N]\oplus\left\|\begin{matrix}* & *\\ 1 & 1\end{matrix}\right\|+\overline{B[2N;2N]}\oplus\|1\ 1\|$$

$$C[2N;2N]=A[2N;2N]\cdot B[2N;2N]$$

上記の符号化手順を次式で表記する．

$$C[2N;2N]=\mathrm{Encode}(A[N;N],B[N;N]) \qquad (20)$$

三枚の二値入力画像の対応画素に対する符号化も，二枚の場合と同様の処理手順で記述できる．

3.2 加　　算

n bit 固定小数点加算に用いる数値データはビットプレーン配置されているものとする．$A_i[N;N]$，$B_i[N;N]$ は入力データの第 i bit を示し，$S_i[N;N]$ は出力データの第 i bit を示す．$C_i[N;N]$ は加

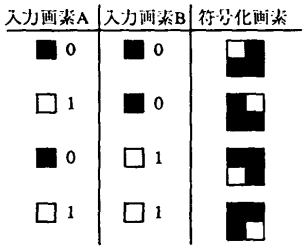


図 7 二枚の二値画像の空間符号化
Fig. 7 Spatial encoding of two binary images.

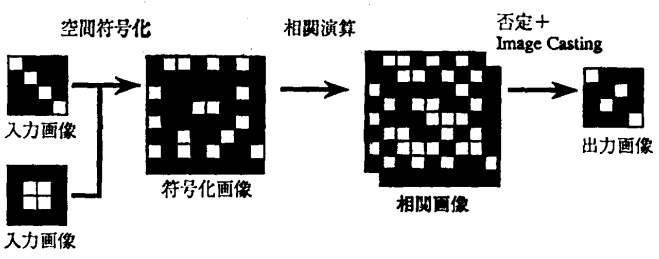


図 8 空間符号化＋相関演算による論理演算
Fig. 8 Logic operation by spatial encoding and correlation

算によるキャリーの第 i bit を示す．$A_n[N;N]$ は最上位ビットを示し，$A_1[N;N]$ は最下位ビットを示す．ただし，$C_1[N;N]$ はすべて0すなわち $C_1[N;N]_{ij}=0$ とする．

次に ILA による固定小数点加算の二つのアルゴリズムを示す．第一のアルゴリズムは，各入力データを空間符号化した後，相関演算 (extended erosion) により論理演算をおこなう方法である．処理手順を**図 8** に示す．原理的には光アレイロジックの演算手順と同一である．第二のアルゴリズムは，各入力データを空間符号化した後，マスク演算により論理演算をおこなう方法である．処理手順を**図 9** に示す．原理的にはスペースバリアント論理演算法の演算手順と同一である．

(第一のアルゴリズム；相関演算による方法)

for $i=1$ to n

$E_i[4N;2N]$
$=\text{Encode}\,(A_i[N;N], B_i[N;N], C_i[N;N])$

$S_i[4N;2N]=\overline{E_i[4N;2N]\otimes\begin{Vmatrix}*&0&0&*\\0&*&*&0\end{Vmatrix}}$

$B_{i+1}[4N;2N]=\overline{E_i[4N;2N]\otimes\begin{Vmatrix}*&*&*&0\\*&0&0&0\end{Vmatrix}}$

$S_i[N;N]=IC(S_i[4N;2N])$

$C_{i+1}[N;N]=IC(C_{i+1}[4N;2N])$

end

(第二のアルゴリズム；マスク演算による方法)

$S[4N;2N]=MI\left(\begin{Vmatrix}0&1&1&0\\1&0&0&1\end{Vmatrix}\right)$

$C[4N;2N]=MI\left(\begin{Vmatrix}0&0&0&1\\0&1&1&1\end{Vmatrix}\right)$

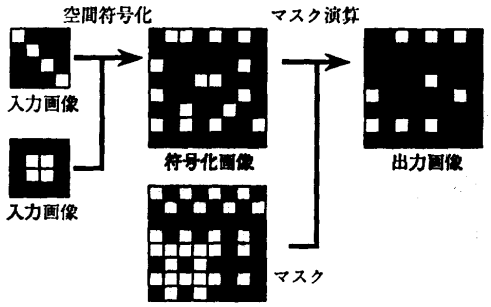


図 9 空間符号化＋マスク演算による論理演算
Fig. 9 Logic operation by spatial encoding and mask operation.

for $i=1$ to n

$E_i[4N;2N]=\text{Encode}\,(A_i[N;N], B_i[N;N], C_i[N;N])$

$S_i[4N;2N]=E_i[4N;2N]\cdot S[4N;2N]$

$C_{i+1}[4N;2N]=E_i[4N;2N]\cdot C[4N;2N]$

$S_i[4N;2N]=\overline{S_i[4N;2N]\otimes\begin{Vmatrix}0&0&0&0\\0&0&0&0\end{Vmatrix}}$

$C_{i+1}[4N;2N]=\overline{C_{i+1}[4N;2N]\otimes\begin{Vmatrix}0&0&0&0\\0&0&0&0\end{Vmatrix}}$

$S_i[N;N]=IC(S_i[4N;2N])$

$C_{i+1}[N;N]=IC(C_{i+1}[4N;2N])$

end

演算回数を比較すると，相関演算による第一のアルゴリズムのほうが，マスク演算による第二のアルゴリズムよりも演算回数が Multiple Imaging 演算二回と AND 演算 $2n$ 回分だけ少ないことがわかる．このことより固定小数点加算に関しては，空間符号化＋相関演算による方法（第一のアルゴリズム）のほうが空間符号化＋マスク演算による方法（第二のアルゴリズム）よりも効率的である．

3.3 乗　　算

電子計算機のほとんどの乗算器では，ブースのアルゴリズム[7]が用いられている．ブースのアルゴリズムは，乗数のビットパターンによって異なる演算を被乗数に対しておこなうため，そのまま SIMD 形式の光並列演算法に適用することは効率的ではない．しかし，前述の ILA によるスペースバリアント論理演算法（空間符号化＋マスク演算）において，マスク画像である $S[4N;2N]$, $C[4N;2N]$ を入力データ（乗数）によって動的に変化させることによって，ブースのアルゴリズムを ILA により効率的に実行可能である．

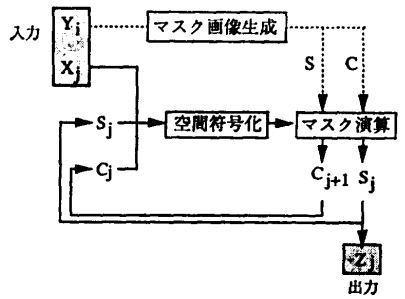


図 10 ILA によるブースのアルゴリズムによる乗算演算
Fig. 10 Multiplication algorithm by Bcoth's algorithm with ILA.

次に ILA におけるブースのアルゴリズムによる固定小数点乗算のアルゴリズムを示す．**図 10** は計算の処理の流れを模式的にあらわしたものである．数値の表現は加算のときと同様にビットプレーン配置を用いる．$X_i[N;N]$，$Y_i[N;N]$ は入力データの第 i bit を示し，$Z_i[N;N]$ は出力データの第 i bit を示す．$S_i[N;N]$，$C_i[N;N]$ は乗算による部分積およびキャリーの第 i bit を示す．ただし，$C_1[N;N]$ はすべて0すなわち $C_1[N;N]_{ij}=0$ とする．乗数 $Y_i[N;N]$ からマスク演算に用いるマスク画像 $S[4N;2N]$，$C[4N;2N]$ を生成し，被乗数 $X_j[N;N]$，部分積 $S_j[N;N]$ およびキャリー $C_j[N;N]$ から符号化画像 $E_j[4N;2N]$ を生成する．符号化画像 $E_j[4N;2N]$ およびマスク画像 $S[4N;2N]$，$C[4N;2N]$ によりマスク演算をおこない，部分積 $S_j[N;N]$ および $C_{j+1}[N;N]$ を得る．以上の演算を i および j を変化させ実行することにより $S_j[N;N]$ に最終的な乗算結果 $Z_j[N;N]$ が得られる．次に具体的なアルゴリズムを示す．

for $i=1$ to n

$X_i[4N;2N]=IC(X_i[N;N])$

$X_{i-1}[4N;2N]=IC(X_{i-1}[N;N])$

$S[4N;2N]$

$=(X_i[4N;2N]\cdot X_{i-1}[4N;2N]$

$+\overline{X_i[4N;2N]}\cdot X_{i-1}[4N;2N])$

$\oplus\left\|\begin{matrix}*&*&1&1\\*&*&1&1\end{matrix}\right\|$

$+(\overline{X_i[4N;2N]}\cdot X_{i-1}[4N;2N]$

$+X_i[4N;2N]\cdot\overline{X_{i-1}[4N;2N]})$

$\oplus\left\|\begin{matrix}*&1&1&*\\1&*&*&1\end{matrix}\right\|$

$C[4N;2N]$

$=(\overline{X_i[4N;2N]}\cdot X_{i-1}[4N;2N])$

$\oplus\left\|\begin{matrix}*&*&*&1\\*&1&1&1\end{matrix}\right\|$

$+(X_i[4N;2N]\cdot\overline{X_{i-1}[4N;2N]})$

$\oplus\left\|\begin{matrix}*&1&*&*\\1&1&*&1\end{matrix}\right\|$

for $j=1$ to n

$E_j[4N;2N]=$

Encode $(Y_j[N;N],S_j[N;N],$

$C_j[N;N])$

$S_j[4N;2N]=E_j[4N;2N]\cdot S[4N;2N]$

$C_j+1[4N;2N]=E_j[4N;2N]\cdot$

$\overline{C[4N;2N]}$

$S_j[4N;2N]=S_j[4N;2N]\otimes\left\|\begin{matrix}0&0&0&0\\0&0&0&0\end{matrix}\right\|$

$C_{j+1}[4N;2N]=C_{j+1}[4N;2N]\otimes\left\|\begin{matrix}0&0&0&0\\0&0&0&0\end{matrix}\right\|$

$S_j[N;N]=IC(S_j[4N;2N])$

$C_{j+1}[N;N]=IC(C_{j+1}[4N;2N])$

end

for $j=1$ to $n-1$

$S_j[N;N]=S_{j+1}[4N;2N]$

end

end

提案したアルゴリズムは従来の光乗算アルゴリズム[8]に比べて二つの数の符号を考慮する必要がないため，二の補数表示数を直接乗算することができる．そのため，SIMD 型の並列計算で効率低下の原因となる前置補数あるいは後置補数演算が不要となり高速化が期待できる．

4. アーキテクチャ

ILA により記述されたアルゴリズムを，光の並列性を活用して効率的に実行するためのアーキテクチャを**図 11** に示す．本アーキテクチャの特徴は，ILA の基本演算をそれぞれ並列に実行する単機能な光演算モジュールと光画像クロスバスイッチ（OPIX)[9] を用いた点にある．

第1の光演算モジュールを単機能とする理由は，複数の演算を一つの光学系で実行しようとすると光学系が複雑になりすぎて実際的ではないためである．なお，この単機能モジュールは，原理的には光学系で実現可能である．例えば画像間の論理演算に関しては，

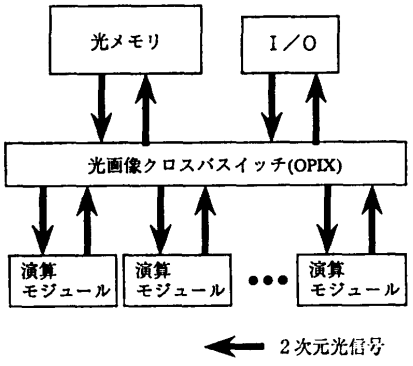


図 11 アーキテクチャ
Fig. 11 System architecture for ILA.

SEED[10]や光サイリスタ[11]等の半導体光論理ゲートを用いた構成が検討されている.

第2のOPIXは,画像を光学的に並列的に変換する非閉塞な空間光スイッチであり,レンズアレイと光半導体素子アレイで構成できる[9]. OPIXには,演算モジュール,画像を並列に読み書きできる光メモリ等が接続されており,入出力データはすべて光信号である.

以上のハードウェアの構成は,現在検討中であり,詳細は別途報告する予定である[12].

5. おわりに

本論文では二次元データの並列処理に適した光演算体系として画像論理代数(ILA)を提案した. ILAでは,新しい概念としてNCPと呼ぶ演算用パターンとその積,和,否定という新しい概念を導入することにより,従来十分な検討がなされていなかった演算用パターンの記述法を確立した. ILAの応用の一例として整数加算,ブースのアルゴリズムによる整数乗算のILAによる記述を示した. 今後,光による具体的なハードウェアおよび画像前処理,LSI CAD等への応用について検討する予定である.

謝辞 日頃からご指導いただいている島田伝送システム研究所長,石尾光通信研究部長に感謝いたします.

参考文献

1) Brenner, K.-H., Huang, A. and Streibl, N.: Digital Optical Computing with Symbolic Substitution, *Appl. Opt.*, Vol. 25, pp. 3054-3060 (1986).
2) Tanida, J. and Ichioka, Y.: OPALS: Optical Parallel Array Logic System, *Appl. Opt.*, Vol. 25, pp. 1565-1570 (1986).
3) Huang, A.: Parallel Algorithms for Optical Digital Computers, Technical Digest, *IEEE Tenth International Optical Computing Conference*, pp. 13-17 (1983).
4) Tanida, J. and Ichioka, Y.: Optical-logic-array Processor Using Shadowgrams. III. Parallel Neighborhood Operations and an Architecture of an Optical Digital-computing System, *J. Opt. Soc. Am.*, Vol. A 2, pp. 1245-1253 (1985).
5) Huang, K.-S., Jenkins, B.K. and Sawchuk, A.A.: Image Algebra Representation of Parallel Optical Binary Arithmetic, *Appl. Opt.*, Vol. 28, pp. 1263-1278 (1989).
6) Yatagai, T.: Optical Space-Variant Logic Gate Array Based on Spatial Encoding Technique, *Opt. Letters*, Vol. 11, pp. 260-262 (1986).
7) Booth, A.D.: A Signed Binary Multiplication Technique, *Q. J. Mech. Appl. Math.*, Vol. 4, p. 236 (1951).
8) Tanida, J., Fukui, M. and Ichioka, Y.: Programming of Optical Array Logic. 2: Numerical Data Processing Based on Pattern Logic, *Appl. Opt.*, Vol. 27, pp. 2931-2939 (1988).
9) 福井,北山:画像クロスバスイッチ(I):構成法と実験結果,応物学会予稿集,10p-ZH-14 (1991).
10) Miller, D.A.B., Chemla, D.S., Damen, T.C., Wood, T.H., Burrus, C.A., Jr., Gossard, A.C. and Wiegmann, W.: The Quantum Well Self-Electrooptic Effect Device: Optoelectronic Bistability and Oscillation, and Self-Linearized Modulation, *IEEE JQE*, Vol. QE-21, pp. 1462-1476 (1985).
11) Taylor, G.W., Mand, R.S., Simmons, J.G. and Cho, A.Y.: Optical Induced Switching in p-channel Double Heterostructure Optoelectronic Switch, *Appl. Phys. Lett.*, Vol. 49, p. 1406 (1986).
12) Fukui, M. and Kitayama, K.: Design Considerations of Optical Image Crossbar Switch (OPIX), *Appl. Opt.*, submitted for publication.

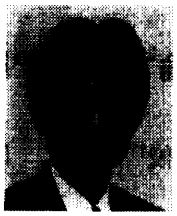
付録 記号一覧

$A[M;N]$	横M画素,縦N画素の二値画像
$A[M;N]_{ij}$	$A[M;N]$の座標(i,j)の画素値
$+$	論理和
$\cdot$	論理積
$A[M;N]\cdot B[M;N]$	画像$A[M;N]$と$B[M;N]$の論理積
$A[M;N]+B[M;N]$	画像$A[M;N]$と$B[M;N]$の論理和
$\overline{A[M;N]}$	画像$A[M;N]$の否定
$\delta_{kl}A[M;N]$	画像$A[M;N]$のシフト変換
$B[R;S]=IC(A[M;N])$	Image Casting変換
$B[R;S]=MI(A[M;N])$	Multiple Imaginng変換
$B[1;1]=TEST(A[M;N])$	TEST変換
$\boldsymbol{R}\wedge\boldsymbol{S}$	NCP RとSの論理積
$\boldsymbol{R}\vee\boldsymbol{S}$	NCP RとSの論理和
$\bar{\boldsymbol{R}}$	NCP Rの否定
$A[M;N]\otimes\boldsymbol{S}$	NCP Sによる画像$A[M;N]$のextended erosion
$A[M;N]\oplus\boldsymbol{S}$	NCP Sによる画像$A[M;N]$のdilation

(平成3年7月30日受付)
(平成3年12月9日採録)

福井 将樹

昭和 62 年大阪大学工学部応用物理卒業．平成元年同大大学院修士課程修了．同年 NTT 伝送システム研究所入社．並列ディジタル光コンピューティングの研究に従事．応用物理学会会員．



北山 研一

昭和 51 年大阪大学工学部通信工学修士課程修了．同年日本電信電話公社茨城電気通信研究所入所．光ファイバケーブル伝送特性，光ファイバ非線形光学の研究に従事．昭和 57～58 年カリフォルニア大学バークレー校客員研究員．半導体レーザ，光集積回路の研究に従事．昭和 60 年研究開発本部．昭和 62 年伝送システム研究所．以来光信号処理，特に光コンピューティング，光ニューラルネットの研究に従事．工学博士．55 年度電子情報通信学会学術奨励賞，60 年度応用物理学会光学論文賞受賞．