Vol. 27 No. 2 情報処理学会論文誌 Feb. 1986

マイクロプロセッサアレイによる2次元ディジタルフィルタの実現†

雛元孝夫†† 中松芳樹†† 小畑正貴†††
辻村敏†† 前川禎男††

画像処理への応用などから，2次元ディジタルフィルタへの関心は高まる一方である．ディジタルフィルタはアナログフィルタに比べ，処理の柔軟性に富む点で優れているが，計算量が膨大になり，処理に時間がかかるのが難点であった．本論文では，並列処理による処理時間の短縮を目的として試作したマイクロプロセッサアレイによる2次元ディジタルフィルタシステムについて述べている．本稿ではまず，対象とするフィルタのモデルを提示し，プロセッサアレイ上への画像データのスケジューリング方法を示す．次に $M\times M$ 台のマイクロプロセッサ(Z-80)で構成したシステムのハードウエア構成と処理動作について詳述する．このシステムはSIMD型であり，処理データへのアドレスを外部のアドレス発生機から供給する点に特徴がある．最後に2×2台の試作機によるフィルタリングの実行例を示し，並列処理の効果について考察する．

1. まえがき

画像認識や画像解析の前処理を行う2次元ディジタルフィルタ技術は，最近の計算機の発達により，急速に進歩している．2次元フィルタは光学的技術によるアナログフィルタに始まったわけであるが，より高精度に，手軽にフィルタリングを行いたいという要求から，ディジタルフィルタへと関心は移ってきた．しかしながらディジタルフィルタでは，処理するデータ量が膨大なため時間がかかるのが問題点となる．そこでディジタルフィルタの高速化についての研究が進められるようになった．

フーリエ変換についてはすでにFFT[1]が開発され，実用化されている．しかし，FFTではすべてのデータの入力が終わらない限り出力はされず，また精度は画像サイズに依存する．これに対し，Roesser形状態空間モデル[2),3)]はこのような欠点がないものであるが，計算量が大きくなる．文献2)はRoesser形状態空間モデルが内包する計算の並列性に着目し，プロセッサアレイによるディジタルフィルタの可能性を提案したものである．本論文では，2)での提案に対していくつかの点で補足，訂正を加え，汎用のマイクロプロセッサを用いて実現したシステムについて述べる．

† Implementation of Two-Dimensional Digital Filters Using a Microprocessor Array by TAKAO HINAMOTO, YOSHIKI NAKAMATSU (Department of Systems Engineering, Faculty of Engineering, Kobe University), MASAKI KOHATA (Department of Electronic Science, Faculty of Science, Okayama University of Science), SATOSHI TSUJIMURA and SADAO Maekawa (Department of Systems Engineering, Faculty of Engineering, Kobe University).

†† 神戸大学工学部システム工学科
††† 岡山理科大学理学部電子理学科

2. フィルタのモデル

ここで対象とするのは次のようなRoesser形状態空間モデル[3)]である．

$$\begin{cases} R(i+1,j)=A_1R(i,j)+A_2S(i,j)+B_1u(i,j) \\ S(i,j+1)=A_3R(i,j)+A_4S(i,j)+B_2u(i,j) \\ y(i,j)=C_1R(i,j)+C_2S(i,j)+Du(i,j) \end{cases} \quad (1)$$

ただし

i,j : 空間座標
$u(i,j)$: 入力ベクトル
$y(i,j)$: 出力ベクトル
$R(i,j)$: 水平方向状態ベクトル
$S(i,j)$: 垂直方向状態ベクトル
A,B,C,D : 適当な大きさの実行列

本システムでは白黒画像を処理するので入出力はスカラである．また R と S にはあらかじめ境界条件が必要となる．

3. 画像のスケジューリング

画素数 $N\times N$ の画像を $M\times M$ 台のプロセッサに割り付けるため，まず画像を**図1**のように $M\times M$ の画像が含まれるような小区画（以下セクションと呼ぶ）に分割する．図において各ます目は画素であり，太枠で囲まれた部分が1セクションである．ます目の数字はその画素を処理するプロセッサの番号であり，例えばプロセッサ(1,1)は各セクションの左上角を担当する．各セクションをアレイ上に配置したプロセッ

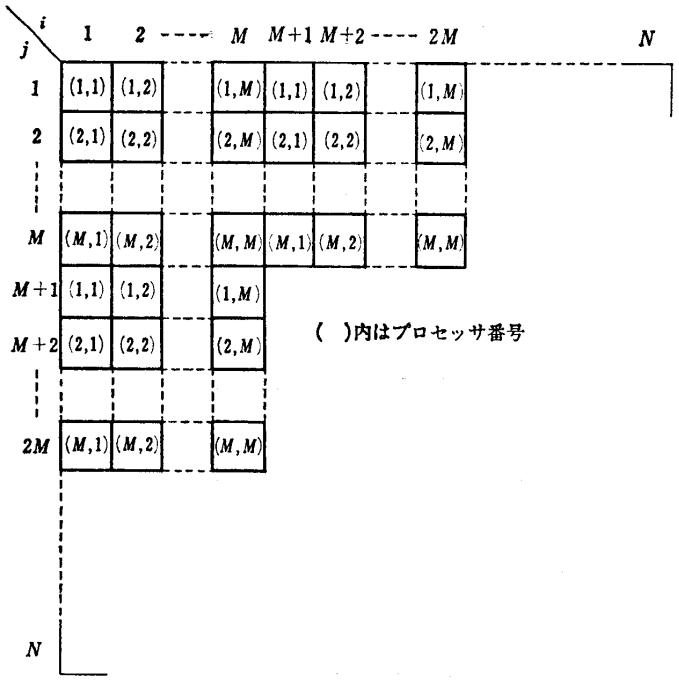


図 1 プロセッサへの画像の割り付け
Fig. 1 Image data allocation to processors.

サで次々にパイプライン的に処理していく．この際，プロセッサ間での状態ベクトルの流れを分断しないように実行順序を決定しなければならない．図2に $M=3$ の場合の例を示す．ます目の数字は計算のサイクルを表している．数字の同じものはその計算サイクルに別々のプロセッサで並列に処理されることを示す．また I, J はセクションの座標である．状態ベクトルは上から下，左から右へと流れていくので，ある画素についての計算サイクルは隣接する左と上の画素についてのサイクルより大きくなる．この理由により，サイクル2,3におけるプロセッサ (1,1) のように手空きとなるプロセッサが最初と最後に発生する．

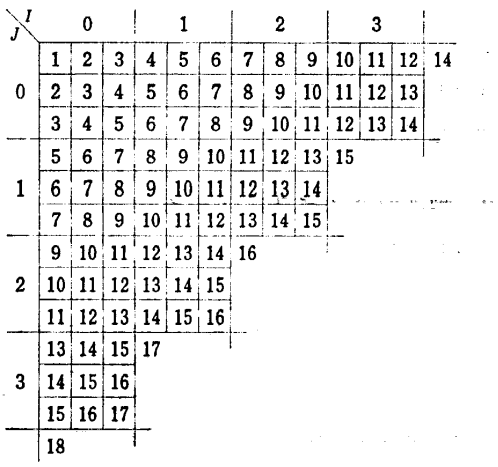


図 2 計算サイクル ($M=3$)
Fig. 2 Execution cycle ($M=3$).

またプロセッサ間での状態ベクトルの転送については，同一セクション内では隣接するプロセッサに対して間をおかず次々に受け渡してやればよい．しかし，セクションの境界では数サイクルを隔てて転送する必要があるため，バッファメモリを介することになる．

さて，図2におけるプロセッサ (1,1) の処理シーケンス（計算するセクションの順序）は次のようになる．

$$(I, J)=(0,0),(0,0)^*,(0,0)^*,(1,0),(0,1),(0,1)^*,(2,0),(1,1),\cdots,(K-1,K-1)$$

ただし $K=N/M$, $M=3$

上のシーケンスにおいて * はそのサイクルが待機（非動作）サイクルであることを示す．

ここでプロセッサ数 M が変わると待機状態のサイクル数が変化する．また，他のプロセッサ (i, j) $(i\neq 1, j\neq 1)$ の処理シーケンスは常にプロセッサ (1,1) から $i+j-2$ サイクルだけ遅れたものとなる．この (I, J) のシーケンスは後述のアドレス発生機が発生する．

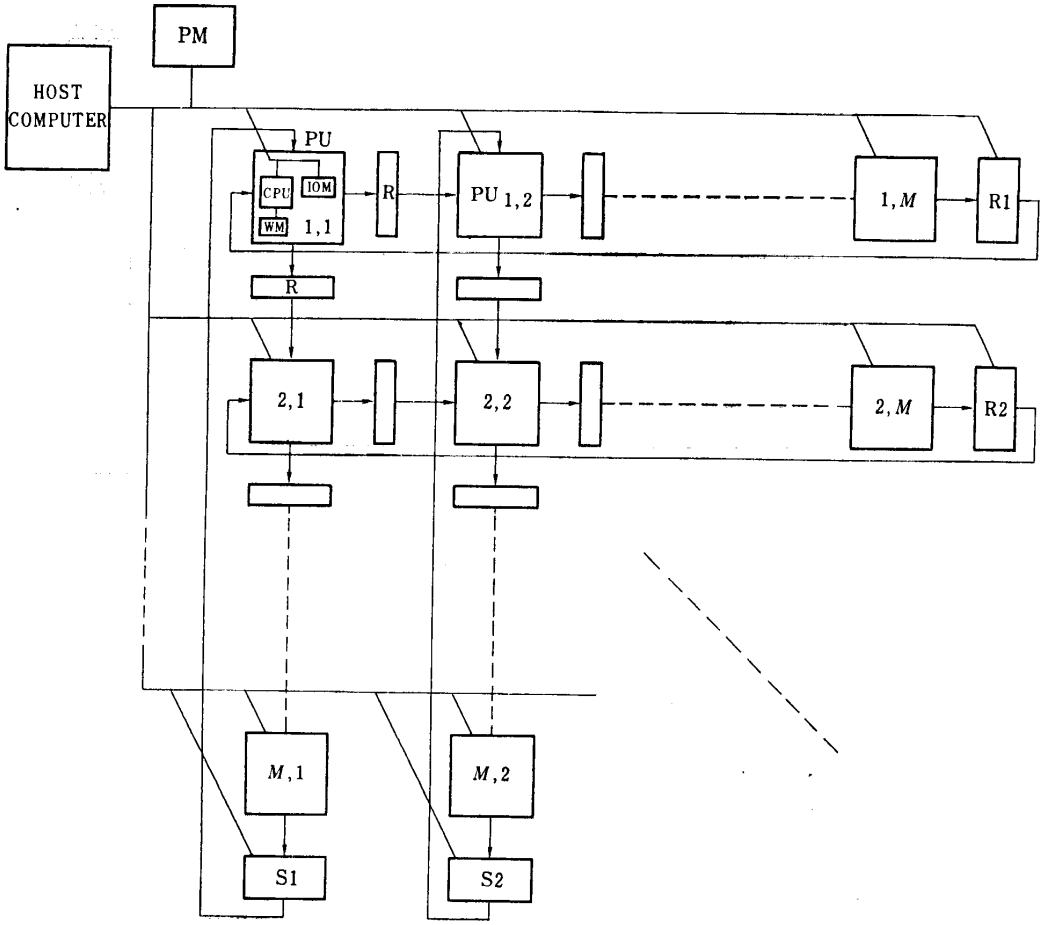
4. ハードウエア構成

$M\times M$ 台のプロセッサ (Z-80) からなるシステムのバス構造を図3に示す．構造は SIMD 型アレイである．アレイを構成するプロセッサのうち，プロセッサ (1,1) を特にマスタプロセッサとする．またシステム全体の制御のため，ホストプロセッサを別に設ける．

4.1 ホストプロセッサ

ホストプロセッサは Z-80 を CPU に持つマイクロコンピュータシステムで，周辺装置として画像入出力装置を持つ．

ホストプロセッサはまずプロセッサアレイにリセットをかけておき，画像データを各入出力メモリに前述した形式で分配しておく．また，状態ベクトルの境界条件を各 R, S メモリへ，フィルタの特性マトリクスとプログラム本体をプログラムメモリへ転送しておく．その後，リセットを解除することにより，プロセッサアレイは一斉に計算を開始する．すべての計算が終了すると，プロセッサアレイ中のマスタプロセッサ



PU: Processing Unit PM: Program Memory IOM: Input Output Memory
WM: Working Memory R,S: 状態ベクトルメモリ R: 状態ベクトルレジスタ

図 3 データバス構成
Fig. 3 Data bus configuration.

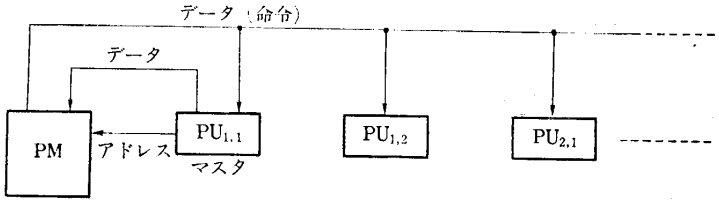


図 4 プログラムメモリへのアクセス
Fig. 4 Access to program memory.

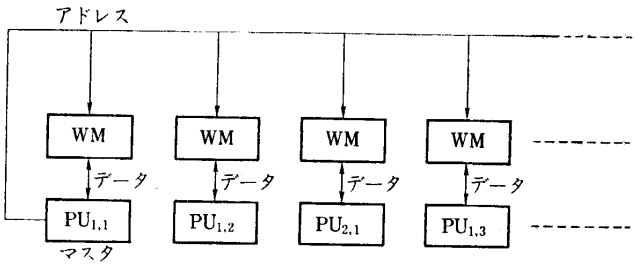


図 5 ワークメモリへのアクセス
Fig. 5 Access to working memory.

は終了フラグを立て，停止する．ホストプロセッサは計算データを収集し，出力画像として再構成する．

4.2 マスタプロセッサ

プロセッサアレイ内では，マスタプロセッサが全体の動作を制御する．本システムは SIMD 型であり，すべてのプロセッサはマスタプロセッサとまったく同じ動作をする．アレイ中の他のプロセッサのアドレス信号および制御信号はまったく使用されない．

4.3 プログラムメモリへのアクセス

プロセッサアレイとプログラムメモリとの間のアドレスとデータの流れを**図 4** に示す．アドレスはマスタプロセッサが出し，プログラムメモリから読み出された命令コードは同時にすべてのプロセッサに配ら

れる．書き込み時は，マスタプロセッサがアドレスを出し，マスタプロセッサからのみデータを書き込むことができる．

4.4 ワークメモリへのアクセス

ワークメモリのアドレスとデータの流れを図5に示す．マスタプロセッサからのアドレスにより，各プロセッサはそれぞれ各自のワークメモリに対して読み書きを行う．ワークメモリはスタックや状態ベクトルの格納に用いる．

4.5 入出力メモリへのアクセス

入出力メモリへのアドレスとデータの流れを図6に示す．下位アドレスは共通でマスタプロセッサが出し，上位アドレスはアドレス発生機が出

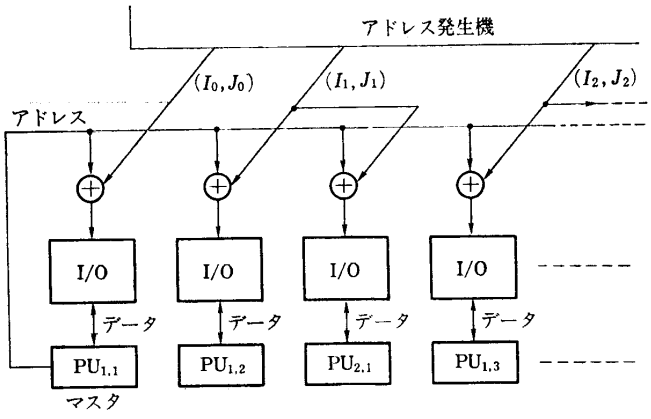


図 6 入出力メモリへのアクセス
Fig. 6 Access to I/O memory.

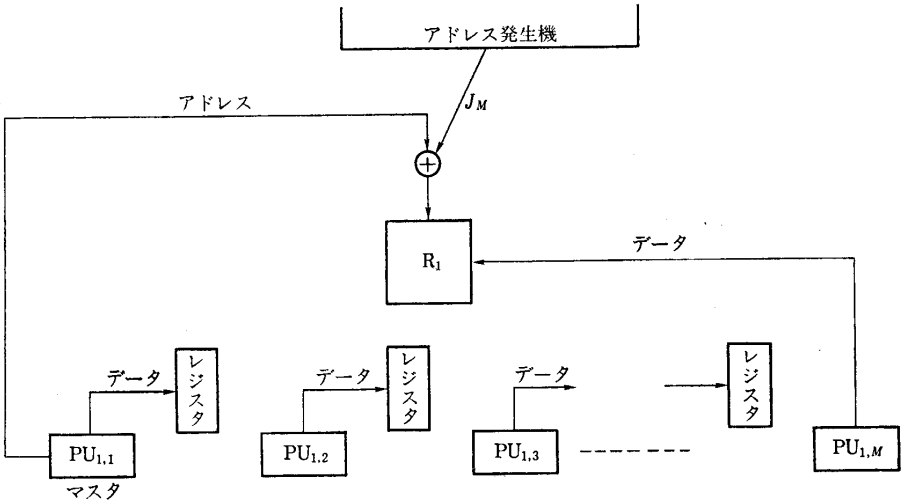


図 7(a) 状態ベクトルメモリ，レジスタへのアクセス（書き込み）
Fig. 7(a) Access to state vector memory/register (data write).

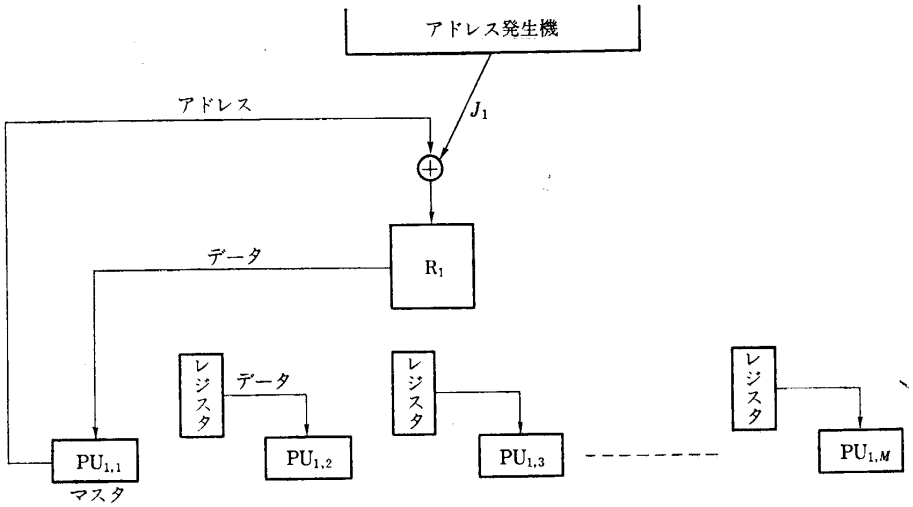


図 7(b) 状態ベクトルメモリ，レジスタへのアクセス（読み出し）
Fig. 7(b) Access to state vector memory/register (data read).

す．上位アドレスは各サイクルにおける各プロセッサの処理セクションの座標であり，前述のシーケンスに従う．

4.6 レジスタおよび状態ベクトルメモリ（R, S メモリ）へのアクセス

状態ベクトル転送時におけるアドレスとデータの流れを図7(a)（書き込み），図7(b)（読み出し）に示す．レジスタは同じセクション内での状態ベクトルの転送に用い，状態ベクトルメモリはセクションの境界での状態ベクトルの転送に用いる．前述したようにセクションの境界では計算サイクルにずれが生じるため，バッファメモリ（状態ベクトルメモリ）により状態ベクトルの転送を適当なサイクルだけ遅らせる必要がある．

状態ベクトルメモリへのアドレスの下位はマスタプロセッサが出し，上位はアドレス発生機が出す．この場合，アドレス発生機の (I, J) 座標のうち R メモリには J 座標を用い，S メモリには I 座標を用いる．

4.7 アドレス発生機

アドレス発生機は計算シーケンスを発生する回路である．各サイクルで，各プロセッサが必要なデータをアクセスするためのアドレスはすべてここから供給される．

アドレス発生機は，マスタプロセッサに対して前述のシーケンスを発生するカウンタと，他のプロセッサにこのシーケンスを一定サイクルだけずらして追従させるための $2M-1$ 段のシフトレジスタ群からなる（図8）．カウンタは次の動作をすることによって必要なシーケンスを発生する．

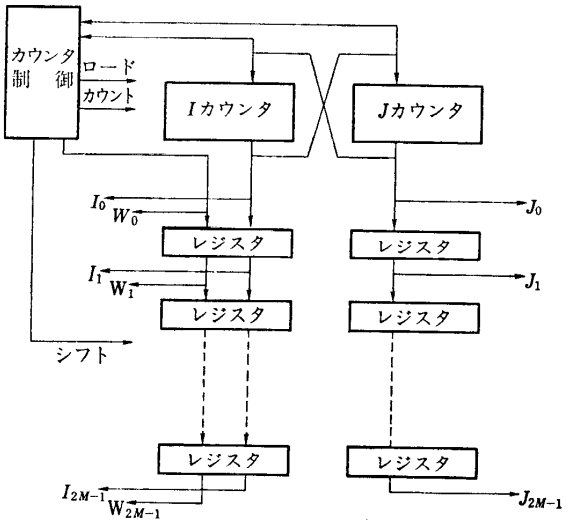


図8 アドレス発生部
Fig. 8 Address generation unit.

i) 初期値 $(0, 0)$

ii) $(a, b) \rightarrow (a-1, b+1)$ $a>0$, $b<k-1$

$(0, b) \rightarrow (b+1, 0)$ $b<k-1$

$(a, k-1) \rightarrow (k-1, a+1)$

値の更新のためのクロックはマスタプロセッサが出す．また，計算の最初と最後で前述の待機サイクルが必要な場合は，マスタプロセッサから禁止信号（図の W_0）が出され，カウンタ値と同様に一定サイクル遅れて他のプロセッサに伝搬する．シフトレジスタ群の第 n 段 $(n=1 \cdots 2M-2)$ には $i+j=n+2$ となるプロセッサ (i, j) が複数台接続される．

プロセッサ数が多くなった場合には各プロセッサへの配線量が大きくなる点が問題となるが，この場合には計算シーケンスを ROM 化して各プロセッサに持たせるようにすればよい．

5. 試作機

試作機として 2×2 のアレイのものを製作した．CPU には Z-80 を用い，クロックは 2 MHz である．メモリはプログラムメモリに 32 k バイト，各入出力メモリに 32 k バイト，各ワークメモリと各状態ベクトルメモリに 2 k バイトずつ実装している．

プログラムはアセンブリ言語によって開発している．画素値は内部では16ビット固定小数点数として計算を行う．乗算などのルーチンは，SIMD 型構造から通常の条件分岐命令が使えないため，プログラミングに工夫を要した．

6. フィルタリング例

図9および図10に低域通過フィルタおよび高域通過フィルタの実行例を示す．ここで低域通過フィルタは

$$A_1=\begin{bmatrix} 0 & -0.836268 \\ 1 & 1.802517 \end{bmatrix},\quad A_2=\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$A_3=10^{-2}\begin{bmatrix} -0.139668 & 1.899036 \\ -0.168818 & 2.492397 \end{bmatrix}$$

$$A_4=\begin{bmatrix} 0 & 1 \\ -0.836272 & 1.802522 \end{bmatrix}$$

$$B_1=\begin{bmatrix} 1 \\ 0 \end{bmatrix},\quad B_2=10^{-2}\begin{bmatrix} 0.521407 \\ 0.700973 \end{bmatrix}$$

$$C_1=10^{-2}[-0.098481 \quad 0.967175]$$

$$C_2=[1 \quad 0],\quad D=0.116807\times10^{-2}$$

の係数行列を持ち，高域通過フィルタは

$$A_1=\begin{bmatrix} 0 & -0.102581 \\ 1 & -1.061383 \end{bmatrix},\quad A_2=\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

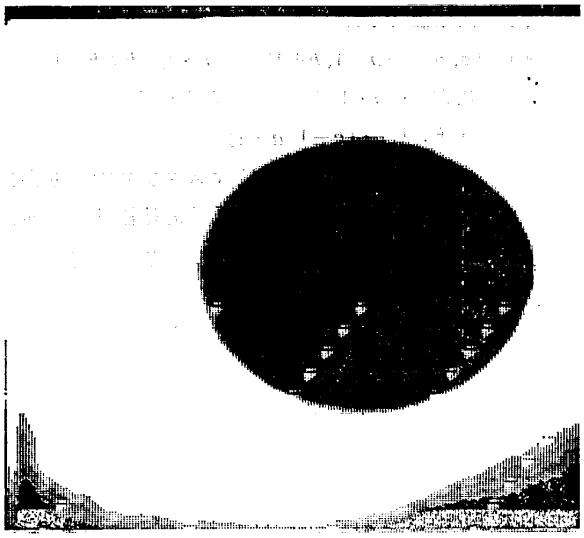


図 9(a) 低域通過フィルタへの入力画像
Fig. 9(a) Input image to the low-pass filter.

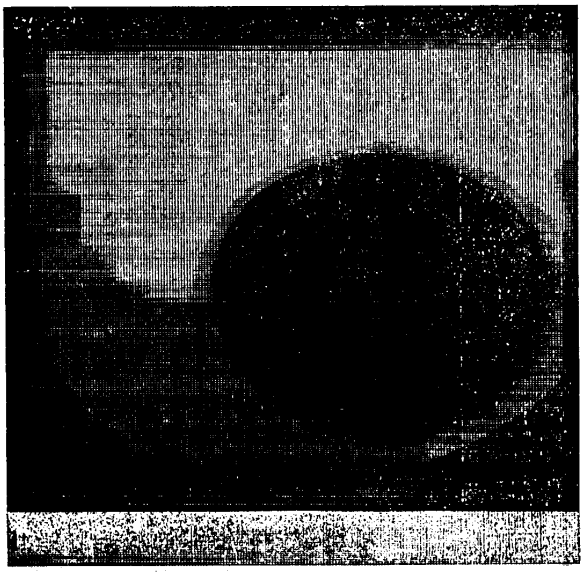


図 9(b) 低域通過フィルタによる出力画像
Fig. 9(b) Output image from the low-pass filter.

$$A_3=\begin{bmatrix}-0.639372 & -0.042977\\ -0.105586 & 0.021151\end{bmatrix}$$

$$A_4=\begin{bmatrix}0 & 1\\ 0.222881 & 0.223849\end{bmatrix}$$

$$B_1=\begin{bmatrix}1\\ 0\end{bmatrix},\quad B_2=\begin{bmatrix}-0.504618\\ 0.577129\end{bmatrix}$$

$$C_1=[0.300436 \quad 0.007664]$$

$$C_2=[1 \quad 0],\quad D=0.473105$$

の係数行列で特定される．A_2 がゼロ行列である構造は，フィルタの伝達関数が分母分離形の場合に相当する．図9(a)のように人為的な雑音ののった原画に

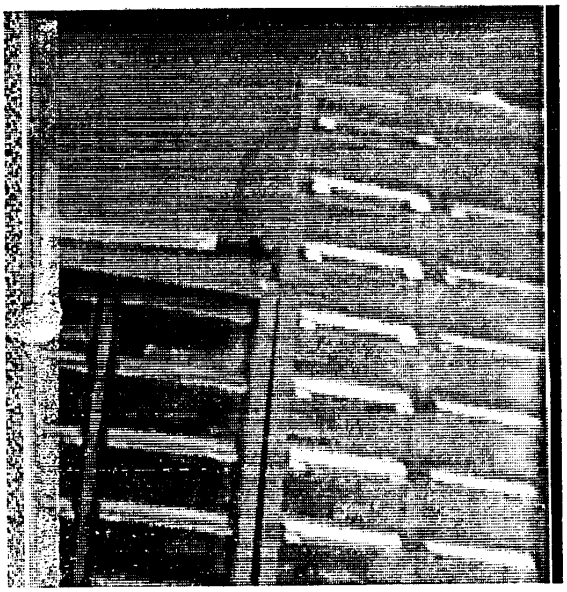


図 10(a) 高域通過フィルタへの入力画像
Fig. 10(a) Input image to the high-pass filter.

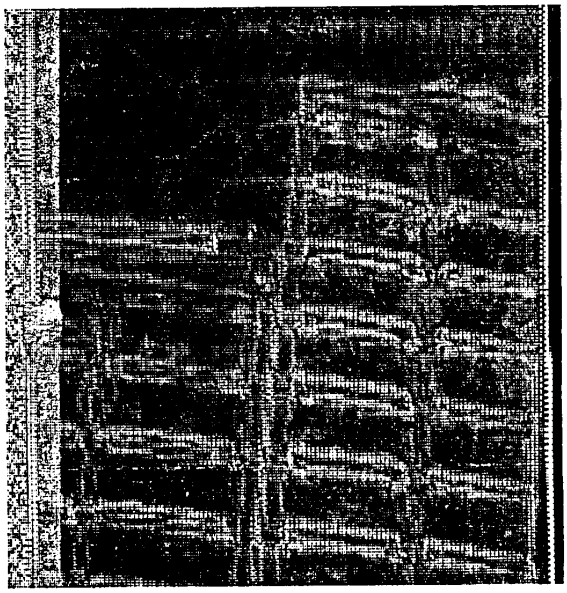


図 10(b) 高域通過フィルタによる出力画像
Fig. 10(b) Output image from the high-pass filter.

本システムによって2次元の低域通過フィルタをかけると図9(b)のように雑音がとれる．

また，図10(a)のような原画に対して高域通過フィルタをかけることにより，図10(b)のように輪郭線を強調した出力画像を得た．これらの結果から本システムがフィルタとして機能していることが確認できる．

7. 並列処理の効果

上の例では256×256画素の画像に対して，2×2の

表 1 実行サイクル数 ($N=256$)
Table 1 Execution cycle ($N=256$).

PU 台数	サイクル数	速度向上比
1^2	65536	1
2^2	16388	3.9990
4^2	4114	15.930
8^2	1094	59.905
16^2	526	124.59
32^2	518	126.52
64^2	514	127.50
128^2	512	128
256^2	511	128.25

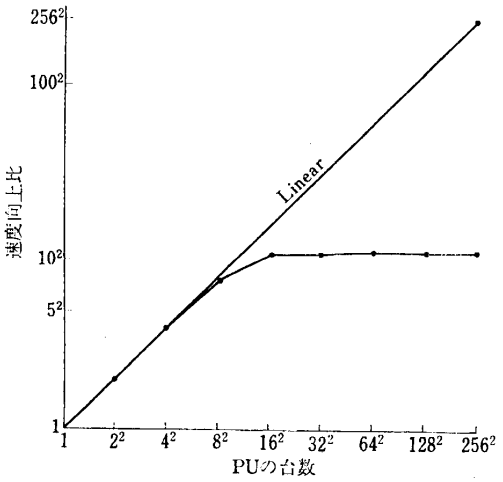


図 11 速度向上比
Fig. 11 Speed up ratio.

構成の試作機によりクロック 2MHz で実行させた．そして，このときの実行時間は13分であった．同じプログラムをホストプロセッサ (Z-80A，クロック 4MHz) 上で実行したところ約2倍の時間を要したことから，4台のプロセッサで約4倍の処理速度が得られたことになり，プロセッサアレイによる並列計算の効果が十分に得られていることが実証できた．しかし試作機は絶対速度が遅く，実時間処理を可能にするには(プロセッサの性能)×(台数)を4桁程度増す必要がある．

一般に画素数を $N\times N$，プロセッサ数を $M\times M$ としたときの実行サイクル数 C は付録により，

$$C=\begin{cases}K^2+M^2+M-2 & N\geq M^2\\ K+2N-2 & N<M^2\end{cases}$$
$$(K=N/M)$$

で与えられる．$N=256$ とし，M を変化させたときのサイクル数と，速度向上比を**表1**および**図11**に示す．これによるとプロセッサが 16×16 台 (すなわち $N=M^2$ のとき) を越えると，ほとんど速度向上が得られないことがわかる．これは待機状態のサイクルが多くなるためである．

しかし処理すべき画像が複数枚ある場合は，現画像での待機サイクル中に次の画像の同一セクションについて計算を進めることができるため，パイプライン処理の効果を生かすことが可能である．

8. むすび

本論文では状態空間モデルの2次元ディジタルフィルタのハードウエアの実現について述べた．状態空間モデルが元来有する高並列計算性を引き出すために，2)での提案をもとにプロセッサアレイとして実現させた．

Z-80 を4台用いた試作機を作成することにより，フィルタとしての機能および並列処理の効果を確認した．またプロセッサ台数に対する総計算サイクル数を導出し，1枚の画像に対しては処理速度の向上に限界があることを示した．このようなシステムは複数の画像に対して十分に多くのプロセッサを用いることにより，並列処理の効果を最もよく引き出すことができる．

なお，本論文で採用したフィルタの構造は状態変数表示によるもので，基本的には巡回形である．これは非巡回形に比べて一般に独立パラメータ数が少なく，フィルタリングにおける乗算回数が少なくて済む．これは処理速度の点で有利である．本巡回形フィルタは，処理例のように空間周波数上の低域あるいは高域等の性質で規定して用いることもできるが，2次元的なカルマンフィルタとしての幅広い応用が大いに期待される．

参考文献

1) Oppenheim, A.V. and Schafer, R.W. (伊達玄 (訳)): ディジタル信号処理，コロナ社，東京 (1978).
2) Roesser, R.P.: Two-Dimensional Microprocessor Pipeline for Image Processing, *IEEE Trans. Comput.*, Vol. C-27, No. 2, pp. 144-156 (1978).
3) Rosser, R.P.: A Discrete State-Space Model for Linear Image Processing, *IEEE Trans. Automat. Contr.*, Vol. AC-20, No. 1, pp. 1-10 (1975).
4) Zilog Inc.: Z 80 Technical Mannual (1977).

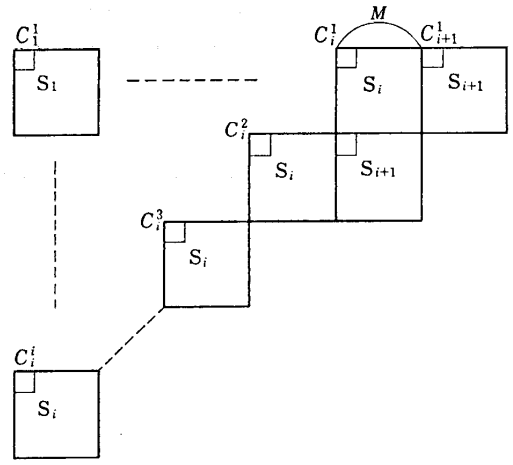


図 A-1 待ちサイクル C_{w1} の導出
Fig. A-1 Derivation of the wait cycle C_{w1}.

付録 計算サイクル数の導出

画素数 $N\times N$, プロセッサ数 $M\times M$ としたときの所要サイクル数を求める. セクション数は $K^2=(N/M)^2$ で与えられる.

心要なサイクル数についてプロセッサ (1, 1) を視点に考えると, まず全セクション数のサイクル数 C_s が必要である. 次に, 開始時と終了時の待機状態のサイクルが必要であり, これを C_w とする. また, 他のプロセッサはプロセッサ (1, 1) に遅延して追従していくので, 最後にこの分だけ遅れる. この遅延サイクル数を C_d とする.

C_s は全セクション数に等しく,

$$C_s=K^2$$

である.

C_w は画像の左上三角部分での C_{w1} と, 右下三角部分での C_{w2} とに分けられる. まず, C_{w1} について考える. **図 A-1** においてプロセッサ (1, 1) がセクション S_i 中の画素 C_i^1 を計算してから, S_{i+1} の C_{i+1}^1 を計算するまでには少なくとも M サイクル間をおかねばならない. この間に計算できるのは $C_i^1\sim C_i^i$ の i 個の画素だけであるから, ここで $M-i$ サイクルの待ちが生じる. $S=1\sim K-1$ の合計として

$$C_{w1}=\sum_{i=1}^{k-1}\max(0, M-i)$$

$$=\begin{cases}M(M-1)/2 & K\geq M\\ (K-1)M-K(K-1)/2 & K<M\end{cases}$$

となる. C_{w2} についても同様で, $C_{w2}=C_{w1}$ となるので,

$$C_w=2C_{w1}=\begin{cases}M(M-1) & K\geq M\\ (K-1)(2M-K) & K<M\end{cases}$$

となる.

C_d は (1セクションの計算サイクル数)-1 であり,

$$C_d=2M-2$$

となる.

したがって全サイクル数 C はこれらの合計として,

$$C=C_s+C_w+C_d$$

$$=\begin{cases}K^2+M^2+M-2 & N\geq M^2\\ K+2N-2 & N<M^2\end{cases}$$

となる.

(昭和60年5月10日受付)
(昭和60年9月19日採録)