

TIPでの画像処理実験

天満 勉・岩下 正雄・溝口 正典

(日本電気(株) C&Cシステム研究所)

1 まえがき

デジタル画像の応用分野が広がるにつれ、画像データの量の多さから、処理高速化が強く要望されている。例えば、ラントラントのMSS 4バンド画像は30Mバイトものデータ量になる。また、画像の処理はマン・マシン系で扱われることが多く、ディスプレイの利用を考えると512x512サイズでR.G.B3面の画像基本処理が2~3秒以内に完了することが望ましい。

これほどTVフレーム時間、或いは数秒で画像基本処理が行えるディスプレイを中心とした画像処理装置が開発された^(1,2)。これらの装置の多くは高速であるが処理機能が固定化される傾向が強く、新しい画像処理アルゴリズムへの対応に難がある。

そこで筆者等は、プログラムの制御が高速処理可能な画像処理ハードウェアシステムTIPを提案、開発してきた⁽³⁾。TIPは複数のモジュールをリニグ状バスで接続した構成で、データ列の処理に必要なモジュールと適切な順序で経由させ、即ち、データの処理手順をソフトウェア的に結合したパイプラインで実現し、データ列間の演算対応制御にデータ駆動制御を取り入れたプロセッサである。

本報告では、TIP-1システム構成を簡単に述べ、速度性能について考察し、2~3の画像処理実験を紹介する。さらに、LSI志向のアーキテクチャであるFIX-TIPの構成、機能、性能を紹介する。

2 TIP-1システム

TIP-1システムの構成を図1に示す。制御部はビント・スライスのマイプロ・プロセッサと主体とした処理部であり、プログラム記述はMPUアセンブラを用いて行なわれる。TIP-1の高速処理部はデータ生成部と演算処理部とで構成され、各部分では複数の処理モジュールがリニグ状バスで接続され同時動作する。この高速処理部のプログラム記述はテンプレート・アセンブラを用いて行なわれる。

2.1 テンプレート・プログラム

TIP-1の高速処理部でフリエ変換処理を行なうためのテンプレート・プログラムを図2に示す。プログラムの各スタートメントは、図1の各処理モジュールでのデータ列間の演算を示しており、入力データ列の名前、演算時のパラメータ、出力データ列の名前が指定される。

2.2 モニタプログラム

モニタプログラムはMPUアセンブラで記述されたプログラムで、ホストとのデータ、プログラムの転送、高速処理部の初期設定及び実行を制御する。図3は図2のテンプレートプログラムと利用しながら二次元FFTを完了させるための不構成プログラムであり、モニタプログラムはこの不構成プログラムを解釈しながら各ノードと逐次処理する。

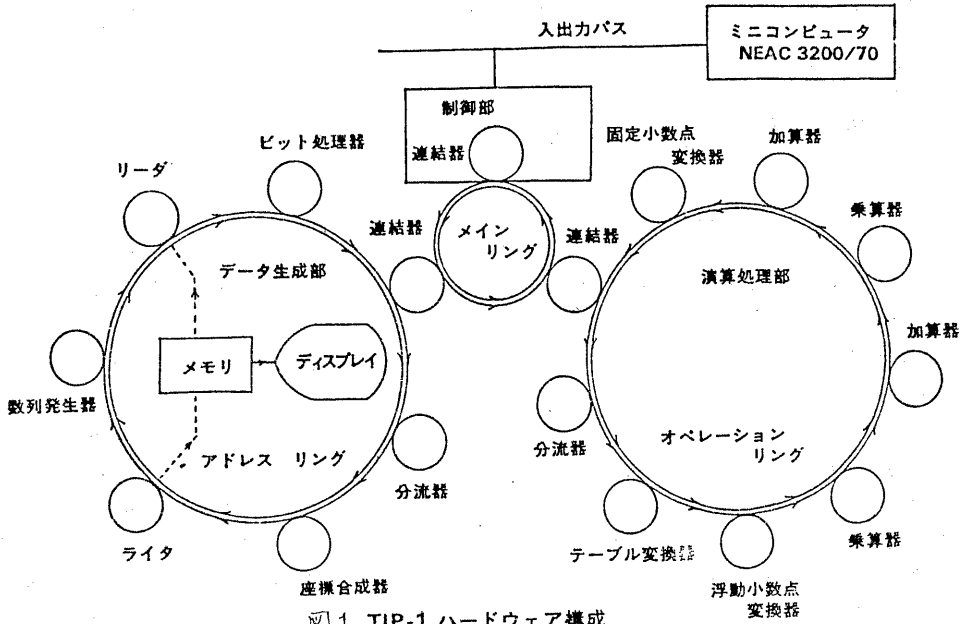


図1 TIP-1 ハードウェア構成

```

=====
%
%      2D-FFT (128*128) X-AXIS
%
%
%
%      URBS : 000800$
%      UIBS : 000800$
%      VRBS : 001800$
%      VIBS : 001800$
%      SCNS : 000000$
%      SCNS' : 000900$
%
%      OUTPUT SRET,TRET
%
%-----
%      SIN - COS WEIGHT SET
%
%      SCNR/SRET = G (SCNS,SCNT /105,1,1/4,0/Z,N,L)
%      SCNA/SCNT = G (SCNR /4,1,4 /1,0/1,N,S)
%      SA = B0 (SCNA /REV,AS,4,FS)
%      CA = D (SA )
%      COS = R (CA /0,880$,W )
%      SIN = R (SA /0,881$,W )
%
%-----
%      INPUT
%
%      FA /TRET = G (SCNS',FSCT /405,805,1/1,1/2,N,L)
%      FSCT = D (SRET )
%      # = D (FSCT ) ; # = FSCT
%      YR,D1,Y1,D2,XR,D3,X1/# = G (FA/7,1,1 /0,0/1,N,S)
%      XRD = R (XR /005,0,W )
%      X1D = R (X1 /305,0,W )
%      YRD = R (YR /405,0,W )
%      Y1D = R (Y1 /C05,0,W )
%      # = FL (D1 /0 )
%      # = FL (D2 /0 )
%      # = FL (D3 /0 )
%
%-----
%      BUTTERFLY PROCESS
%
%      RR = MO (YRD,COS /V,405 )
%      IR = M1 (Y1D,COS /V,405 )
%      RI = M1 (YRD,SIN /V,405 )
%      II = MO (Y1D,SIN /V,405 )
%      WR = AO (RR,II /Q,105 /SUB)
%      WI = A1 (IR,RI /Q,105 /ADD)
%      UR = AO (XRD,WR /Q,105 /ADD)
%      VR = AO (XRD,WR /Q,105 /SUB)
%      UI = AO (X1D,WI /Q,105 /ADD)
%      VI = AO (X1D,WI /Q,105 /SUB)
%
%-----
%      OUTPUT
%
%      W (UR,URBS /V,105 /20005,1,405 /W,VA,A )
%      W (UI,UIBS /V,105 /20005,1,405 /W,VA,A )
%      W (VR,VRBS /V,105 /20005,1,405 /W,VA,A )
%      W (VI,VIBS /V,105 /20005,1,405 /W,VA,A )
%
%      END

```

図2 テンプレート・プログラム

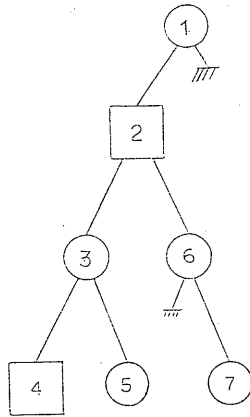


図3 木構造プログラム

1. Template program for FFT preparation.
2. Template program initialization in node 3.
3. Template program for FFT stage.
4. Template program modification in node 3.
5. FFT execution for x- and y-direction.
6. Data replacement in x-direction.
7. Data replacement in y-direction.

3 TIP-1の速度性能

TIP-1の速度性能はパイプライン方式での速度性能をベースに考慮でき⁽³⁾が、TIP-1に特有の問題としてリング状データバスでのデータ転送能力がある。

図4はTIP-1の処理スケジュールに共通なバス・インタフェース部を示している。リング状バスのデータ転送は50ns毎に可

能であるが、処理モジュールは 100ms 或いは 150ms 毎にデータを受けとります。このため、同一モジュールに連続してデータが到着すると、そのモジュールはデータをとり込むまでリングホバスの転送を停止させる。この停止状態によりバス転送能力の低下が TTP-1 の処理性能を下げている。

図5は 100ms モジュール、150ms モ

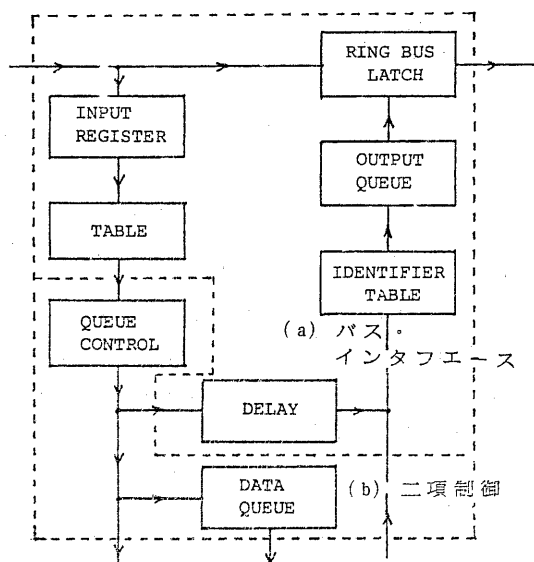
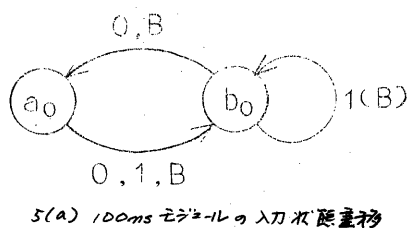


図4 バス・インタフェース



5(a) 100ms モジュールの入力状態遷移

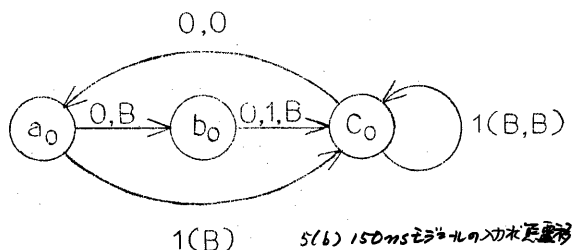


図5 入力レジスタの状態遷移

ジュールでの入力レジスタの状態遷移を示している。例えば、図5(a)は 100ms モジュールの入力状態遷移を示しており、100ms の 1 周期の最初の 50ms (状態 a0) ではデータが到着しようとそうであるがリングホバスの転送停止の状態にあると状態 b0 に遷移する。その後の 50ms (状態 b0) ではデータが到着すると次のデータ転送を停止させる。それ以外では状態 a0 に遷移する。図5をもとに、データの到着率 p 、転送停止の発生率 a_0 、 b_0 、 c_0 を用いると定常状態では (1) 式が成立する。

$$\begin{bmatrix} a_0 \\ b_0 \end{bmatrix} = \begin{bmatrix} 0 & 1-(1-b_0)p \\ 1 & (1-b_0)p \end{bmatrix} \begin{bmatrix} a_0 \\ b_0 \end{bmatrix} \quad (1)$$

$$\begin{bmatrix} a_0 \\ b_0 \\ c_0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & g+CbP \\ g+AbP & 0 & 0 \\ (1-a_0)p & 1 & (1-c_0)p \end{bmatrix} \begin{bmatrix} a_0 \\ b_0 \\ c_0 \end{bmatrix}$$

但し、 $g = (1-p)$ であり、例えば b_0 は 100ms モジュールの b_0 の状態での転送停止発生率を示している。

(1) 式を解くと (2) 式が得られる。

$$P_0(i, 2, s) = \begin{cases} (1-b_0)p & : s = a_0 \\ 0 & : s = b_0 \end{cases} \quad (2)$$

$$P_0(i, 3, s) = \begin{cases} (1-c_0)p & : s = a_0 \\ (1-c_0)p + (1-a_0)(1-(1-c_0)p) & : s = b_0 \\ 0 & : s = c_0 \end{cases}$$

但し $P_0(i, j, s)$ で i はモジュール番号、 $j=2$ は 100ms モジュール、 $j=3$ は 150ms モジュール、 s は転送カプフルを示している。

図2の FFT 処理のバタフライ処理部の流量図は図6のようになる。図6で処理モジュールの上に表示した数値はそのモジュールへのデータ入力数、モジ

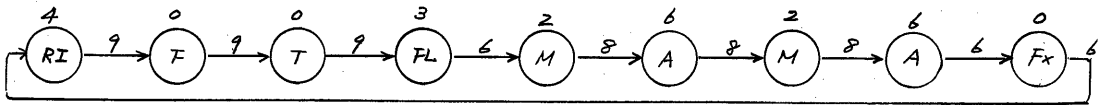


図6. FFT バタフライ処理のデータ流量図

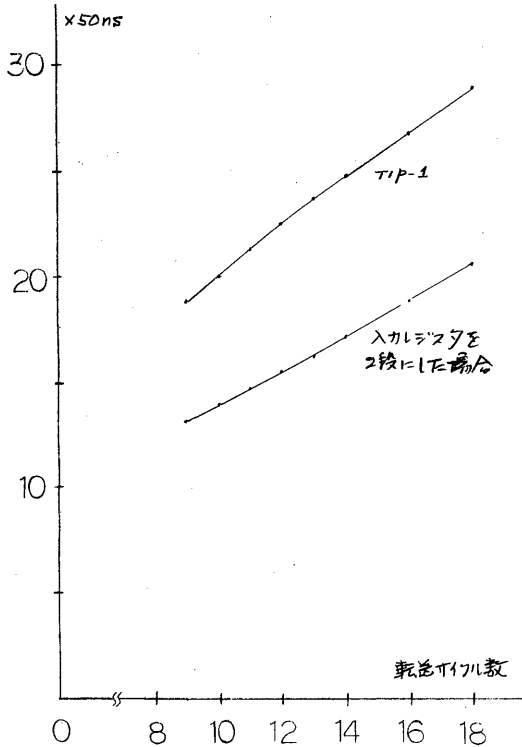


図7. バタフライ処理に対するバス転送能力

モデル上に示した数は転送されねばならないデータ数を示している。

(2)式に図6を適用し定常状態でのバス転送時間を計算した結果を図7に示した。TIP-1ではバタフライ処理が最高950ms程度で処理できることを示しており、空のデータが1つ余分に転送された場合1050msに低下する。また入力レジスタが2段ある場合を仮定してモデル解析を行なったバタフライ処理は最高速700ms程度で動作しうま予想される。

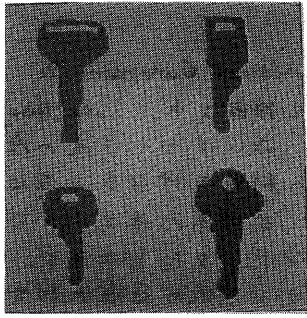
TIP-1とて図2のテンプレートプロ

グラムを実行した時のバタフライ処理時間を計測した結果は1050msであり、空データの転送も観察されることからモデル解析が実際とかなり近いことが示された。

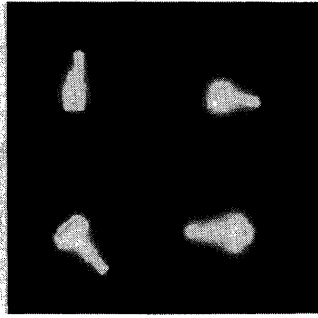
4. TIP-1での画像処理実験

TIP-1の高速性と汎用性を調べため、画像の基本的な処理、アフィン変換、レンズ系座標変換、マスク処理、FFT処理を行ない、組み合わせ処理として画像照合処理を行なった。256x256ピクセルの画像のアフィン変換に約30ms、レンズ系座標変換に約60ms、マスク処理に約130msが計測され、128x128ピクセルの複素フーリエ変換にポイントリバーサル転置も含めて約150msが計測された。

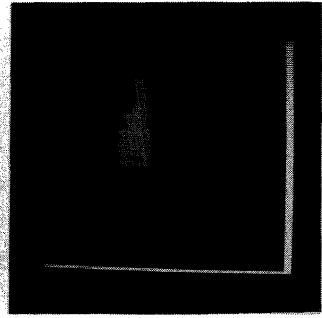
図8は画像処理実験時の各画像を示している。図8(a)~(f)は画像照合実験を示しており(a)は標準画像、(b)は入力画像、(c)は二値化後の画像、(d)~(f)は画像の選択、平行移動量、回転角算出後のデモンストレーション表示である。この照合実験でTIP-1は入力画像を受け取った後、マスク処理、二値化、重心位置検出、重心を中心として放射方向のヒストグラム計算、ヒストグラム間の相関計算と最大値検出を不構造化プログラムに交行行なう。図8(g)~(i)はレンズ系座標変換処理を示しており、図8(j)は図8(i)を平滑化した画像、図8(k)は図8(j)をアフィン変換した画像である。



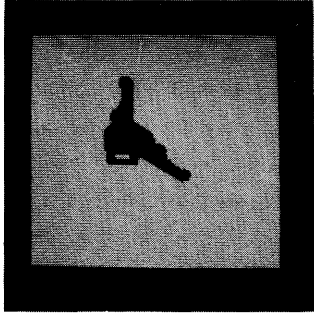
(a)



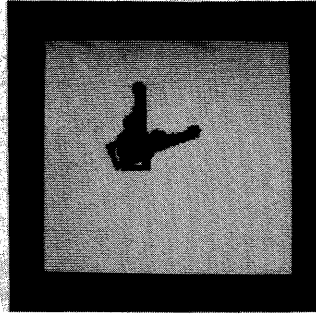
(b)



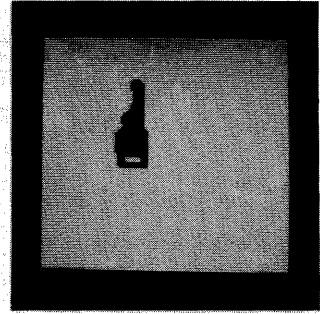
(c)



(d)



(e)



(f)



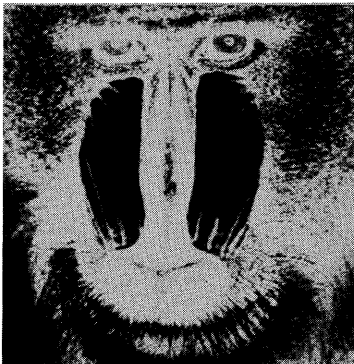
(g)



(h)



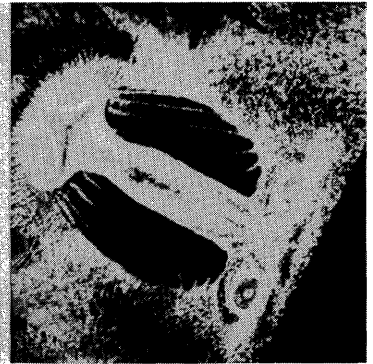
(i)



(j)



(k)



(l)

图 5 图像处理实验例

5 モジコラ型 TIP

TIP-1 システムは各モジュールが単機能モジュールとして実現されており、開発工数、メンテナンスの意味で同類がある。加えて低価格化、小型化には最近の VLSI 技術が必要であり、モジコラ型が適している。このため、TIP-1 の各モジュールの機能を統合し簡素化した構成のモジュール FIX-TIP を試作し、性能評価を進めている。

5.1 FIX-TIP の構成

図9に FIX-TIP の構成を示す。Input Controller, Output Controller, Output Queue が TIP-1 のバスインタフェースに対応する。FIX-TIP は内部にリニアのバイプロセッシング系 (Link Table, Function Table, Data Memory, Queue, Processing Unit) をもち、外部リニアバスでのデータ転送量を軽減している。

Input Controller からデータが入力されると、Link Table に転送され、Function Table のアドレスと新たな ID 番号とが与えられる。Function Table から読み

出された内容は Address Generator & Flow Controller で解釈され、Data Memory の利用を制御し、或いはデータ流の分岐、カウント等の制御を行ったりと共に Processing Unit で演算処理されるための命令コードをデータに付加する。

Queue は内部リニア系の処理を初歩よく動作させるための緩衝用バッファであり、2種類の Queue で構成されている。1つは演算処理が行われるデータのための Queue であり、他の1つはデータ生成が行われるデータに対する Queue である。この2つの Queue から Processing Unit へのデータ転送の優先順位を動的に制御することによって内部リニア上のデータ量を制御している。

5.2 アセンブリ

図10に TIP バージョフアセンブリを用いて記述したアセンブリプログラムの一部を示す。アセンブリでは、FIX-TIP への入力データ名、FIX-TIP からの出力データ名を記述した後、LINK 文、FUNCTION 文、MEMORY 文に

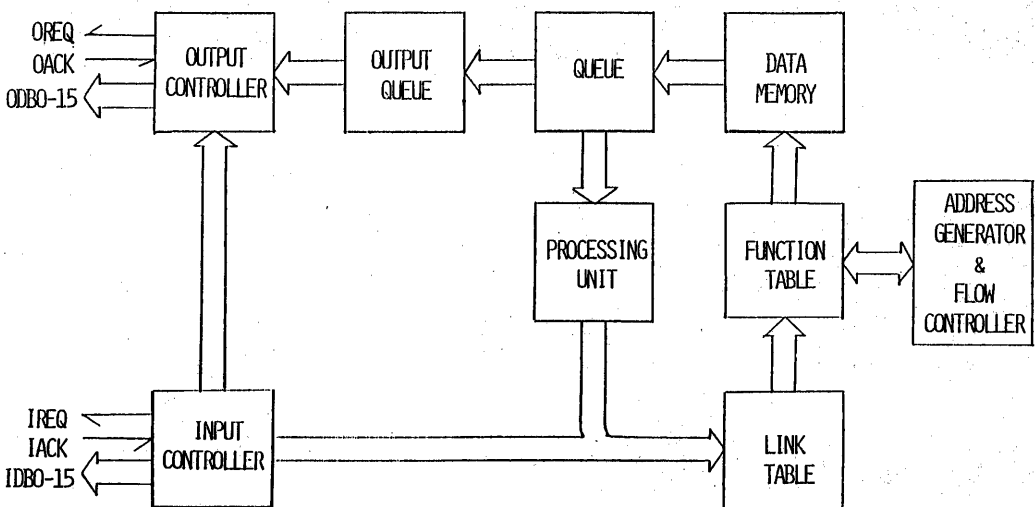


図9 FIX-TIP の構成

より、処理の流れと処理の内容とを記述する。LINK文はデータの接続関係を示し処理の流れを明らかにすると共にどのFUNCTION文、MEMORY文を利用するかを示す。FUNCTION文はAddress Generator & Flow Controller での動作或いは Processing Unit での動作を指定する。MEMORY文は Data Memory の利用先頭アドレスと利用語長を定義する。

Address Generator & Flow Controller での動作のうち Data Memory に関するものとして、変数名毎のデータ駆動制御、数値列とマイクロプログラムに対応する制御、作業用バンプ、テーブルルックアップ等の機能があり、データ列を分岐、計数するものとして幾つかの方法が用意されている。

```

PROGRAM AFIN;
*
IFROMCPU A11,A12,A13,A21,A22,A23,SIZ,C;
OINTOCPU STOP;
OINTOMEM AL,DAT,ADR;
*
LINK GC1,RT1,STP = G1(RT1,RTN);
LINK STOP = ED(STP);
LINK Y1,Y2,Y3,Y4 = C1(GC1,NUL);
LINK GC2,RT2,RTN = G2(RT2,Y1);
LINK X1,X2,X3 = C2(GC2,NUL);
LINK SZ1,SZ2 = C4(SIZ,NUL);
LINK M1 = ML(Y2,SZ1);
LINK A1 = AD(M1,X1);
LINK DAT = RD(A1);
LINK CP1,CP2 = C3(C,NUL);

FUNCTION ML=PU(1,0,0,0,0,MPY);
FUNCTION ML=DTQ(0,1);
MEMORY ML=QUEUE;
FUNCTION AD=PU(1,0,0,0,0,PLS);
FUNCTION AD=DTQ(1,1);
MEMORY AD=QUEUE;
FUNCTION ED=OT(0,0,0,0,0);
FUNCTION RD=OT(0,0,0,0,4);
FUNCTION WT=OT(0,0,0,0,5);
*
READL A1;
WRITED DAT;
WRITEL ADR;
*
END;

```

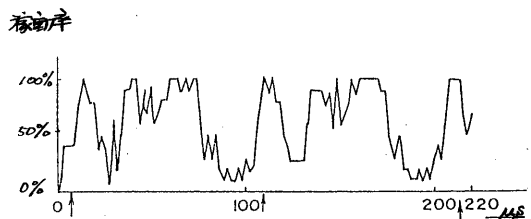
図10 アフィン変換プログラム (部分)

また、Processing Unit は 16 ビットの ALU, 16 ビットの乗算器, ビットリバーシ回路, エンコーダ回路で構成され、Logical 演算, Arithmetic 演算, Shift 処理, 比較演算の他、数値列発生, コピー, ビット処理等が行なわれる。

Address Generator & Flow Controller の命令と Processing Unit の命令を組み合わせてからデータ駆動制御でデータ処理を行わせることにより Tip-1 では実現出来たが、IF THEN ~ ELSE の処理、倍精度の処理等が可能となった。

5.3 FIX-TIP の速度性能

FIX-TIP での処理時間は Link Table に入力される各変数名のデータ量の総和で見積りことが出来る。図 11 は 1024 x 1024 サイズの 2 値画像の平滑化処理を ACOS 900 でシミュレーションした結果である。横軸は実行時間、縦軸は Link Table に入力される有効データの量を示している。FIX-TIP の内部パイプラインリングが完全につまった場合 1024 x 1024 サイズの平滑化処理は約 4 秒で完了する。実際には、無効データが 37.2% Link Table に流れ実行時間は 6.4 秒になる。最大の理由はアドレス列の発生が行なわれ、外部の画像メモリをアクセスし、データ処理後



1024x1024 2値画像の平滑化: 6.4 s
平均稼働率: 62.8%

図11 平滑化処理の稼働率

基本処理	画像メモリのアクセス回数		処理データ数	実行サイクル	稼働率	実行時間 ms
	読み出し	書き込み				
回転角検出	24,740	720	690,000	1,048,000	66.0%	209.6
アフィン変換	65,536	4096	1,700,000	1,899,500	89.5%	379.9
-微分計算	4096	0	61,000	98,000	62.2%	19.6
X方向アフィン	4096	256	226,500	348,000	65.1%	69.6
Y方向アフィン	4096	256	66,500	99,000	67.2%	19.8

表1 FIX-TIPでの基本画像処理に対する処理時間

画像メモリに再び格納される周期の間にFIX-TIPでのデータの流しが遅くなるためと考えられる。

256x256ピクセルの画像に対し、画像照合に必要な基本処理実行時の計測値を表1に示した。表1で1実行サイクルは200msに対応している。

6 あとがき

新しい画像処理アルゴリズムにも適用できる高速画像処理ハードウェアの基本評価を行なった。その結果、高速性、汎用性についても満足する結果が得られた。実行時、リングバスの転送能力がバス転送帯域幅の発せにより低下することが分かったが、確率モデルを提案、定常解を求めることで実行時に近い転送能力が計算されることを示した。これによりプログラムの実行時の処理速度があらかじめ近似的に予測できる。また、TIP-1システムのバスインタフェース部の入力レジスタが現在1レベルであるが2レベルにすることで性能が大幅に改善されることを予想される。

TIP-1をベースにLSI志向のTIPアーキテクチャFIX-TIPを示した。現在、FIX-TIPのシミュレーション、試作機、アセンブリが完了し、性能評価、プログラムの開発を進めている。

複教台のFIX-TIPを容易に利用出来る言語の開発を早急に行ない試作完了しているFIX-TIP3台のTIP-2システム上でデータの流しを計測すると共に性能評価を行う予定です。

[謝辞]

日頃御指導頂く周辺機器研究部花本部長、首藤課長、TIP-1のソフトウェア開発を分担して頂く入沢玲子様、FIX-TIPの性能評価を進めて下さる下野博子様に感謝します

[参考文献]

- (1) Y. Sorimachi et al., "Economical Landsat Image Analysis (ELIAS)," NEC R&D, No.60, pp 26-32, Jan. 1981
- (2) 木戸公, 「画像処理ハードウェア」情報処理 Vol 21, No. 6, June 1980
- (3) T. Tamma et al., "Template-controlled Image Processor TIP-1 Performance Evaluation," Proc. of CVPR '83
- (4) M. Iwashita et al., "Modular Dataflow Image Processor," Proc. of Compcon '83 Spring, pp. 464-467, 1983