

## TIP-1での画像処理

溝口 正典 天満 勉

日本電気(株) C&amp;Cシステム研究所

## 1 まえがき

TIP-1システムは複数個のモジュールをソフトウェア的に結合し、パイプライン路を設定することで、モジュールをパイプライン的に並行動作させ、高速処理を実現させるデータ駆動制御のマルチプロセッサ部と、低速ではあるが処理に柔軟に対処できるマイクロプログラム制御の Control Unit 部 (CU部) とで構成される。CU部は逐次実行方式であり、MPUアセンブラでマイクロプログラムを記述する。一方、マルチプロセッサ部のパイプライン路を決定するテンプレートプログラムは、テンプレートアセンブラで記述されたものが各モジュールに分散され、データ駆動方式で実行される。

テンプレートプログラムの設定と処理開始の制御はCU部が担当している。

画像アプリケーション処理はいくつかの画像基本処理の集まりとして構成されており、各基本テンプレートプログラムを順次実行したり、条件判断やホストミニコンとのデータ転送等のMPU処理ルーチンとの組み合わせによって実行される。

本報告では、CU部において、基本テンプレートプログラムとMPU処理ルーチンとの実行を制御するモータプログラムを設計・作成し、実際に画像処理実験を行った結果について述べる。

## 2 TIP-1のプログラミング

## 2.1 テンプレートプログラム

マルチプロセッサ部のテンプレートプログラムは関数形式のテンプレートアセンブラ<sup>(1)</sup>によって記述される。TIP-1のシステム構成<sup>(2)</sup>を図1に示す。関数は各モジュール毎に用意されており、1つのステートメントで、入力データ名、各演算制御パラメータ、出力データ名等を記述する。関数形式ステートメントの例を表1に示す。またプログラム例を図2に示す。

アSEMBルされたテンプレートプログラムは、CU部のテーブルメモリにストアされる。プログラムの実行は、CU部から前記テンプレートを初期セットトークンとして送出し、Generatorでの数列データ生成のトリガとなる起動トークンを出力することで開始され、終了トークンの入力で終了する。

## 2.2 MPUプログラム

CU部は9個のレジスタ、4個の演算エレメント、メモリ等を2系統の読み出しバスと、2系統の書き込みバスで接続した構成である。(図3)

MPUアセンブラのステートメント<sup>(3)</sup>は、2系統のデータ転送の source, destination, operation parameter 等が記述される。MPUアセンブラの命令を表2に示す。

マルチプロセッサ部は、ホストミニコンピュータとのデータの授受は、Input Register, Output Register との Read, Write によって行われる。

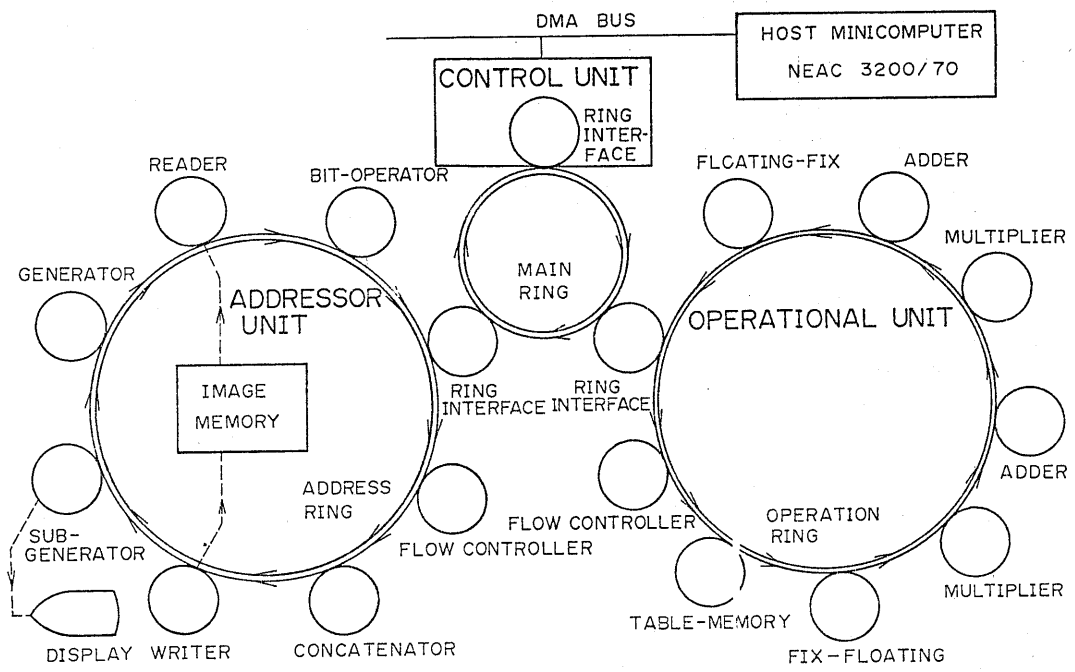


図1 TIP-1 システム 構成図

モジュール	関数形
Multiplier	OUT = MQ ( INe, INo / 2項モード指定, キーの深さ )
Adder	OUT = AQ ( INe, INo / 2項モード指定, キーの深さ / 演算コード )
Generator	OUT1, ..., OUTn / DEAD = G ( INst, INcs, INst, INcs / x座標の y の初期値, x座標の y の初期値生成, DEAD, 座標座標 / 座標座標, 偏置値, モード )
Writer	W ( INd, INa / 2項モード指定, 深さ / オブジェクト増分, 画面数 / アドレスサイズ, 定数値生成, モード )
Reader	OUT = R ( IN / x座標アドレス, y座標アドレス / アドレス )
Bit-Operater	OUT = BO ( IN / ファンクションコード, シフト数, マスクビット, マスク幅 )

表1 テンプレート・アセンブラ・ステートメント例

```

Z ID-FFT (128*128)
Z
URBS : 080000#
UIBS : 080080#
VRBS : 080001#
VIBS : 080081#
SCHS : 000000#
Z
OUTPUT TRET
Z
Z SIN, COS WEIGHT ADDRESS SET
Z
SCNR/SRET = G (SCNS, SCNT /F#,1,1 /4,0 /2,N,L)
SCNA/SCNT = G (SCNR /4,1,4 /1,0 /1,N,S)
SA = B (SCNA /REV;6,9,F#)
CA = D (SA)
COS = R (CA /0,800#,M)
SIN = R (SA /0,801#,M)
Z
Z INPUT
Z
FSCR/TRET = G (SRET, FSCT /1,80#,1 /0,1 /2,N,L)
FA/FSCT = G (FSCR, VI /40#,1,1 /1,0 /1,N,L)
YR, YI, XR, XI = G (FA /4,1,1 /0,0 /1,N,S)
XRD = R (XR /00#,0,M)
XID = R (XI /80#,0,M)
YRD = R (YR /40#,0,M)
YID = R (YI /80#,0,M)
Z
Z BUTTERFLY PROCESS
Z
RR = M0 (YRD, COS /V,40#)
IR = M1 (YID, COS /V,40#)
RI = M1 (YRD, SIN /V,40#)
II = M0 (YID, SIN /V,40#)
Z
WR = A0 (RR, II /0,10# /SUB)
WI = A1 (IR, RI /0,10# /ADD)
Z
UR = A1 (XRD, WR /0,10# /ADD)
VR = A0 (XRD, WR /0,10# /SUB)
UI = A0 (XID, WI /0,10# /ADD)
VI = A1 (XID, WI /0,10# /SUB)
Z
Z OUTPUT
Z
W (UR, URBS /V,10# /2,1000# 40# /N,VA,A)
W (UI, UIBS /V,10# /2,1000# 40# /N,VA,A)
W (VR, VRBS /V,10# /2,1000# 40# /N,VA,A)
W (VI, VIBS /V,10# /2,1000# 40# /N,VA,A)
Z
END

```

図2 テンプレートプログラム例1 (-次元 FFT バタフライ1ステージ分)

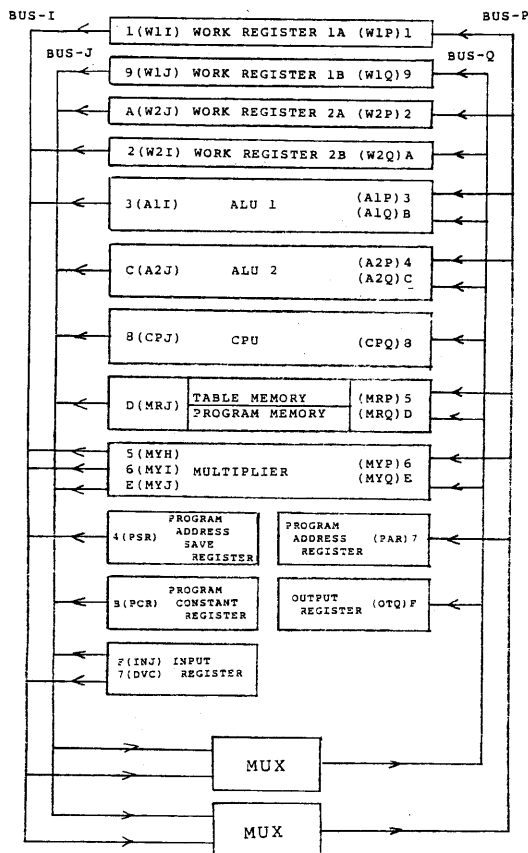


図3 CPU部(MPU)の構成

区分	命令	パラメータ
CNT	HLT	
	NOP	F <sub>1</sub> , T <sub>1</sub> , FNC
TRS	TRS	F <sub>1</sub> , T <sub>1</sub> , FNC, F <sub>2</sub> , T <sub>2</sub> , T <sub>3</sub> , T <sub>4</sub>
CPU	ADD, SUB	F <sub>1</sub> , T <sub>1</sub> , FNC, load, store, shift
	RVS, RVD	
	OR, AND	
	MSK, XOR	
	INC, DEC	
CON	SET	F <sub>1</sub> , T <sub>1</sub> , FNC, CONSTANT
	ADR	F <sub>1</sub> , T <sub>1</sub> , FNC, ADDRESS
JMP	JMP, JST	F <sub>1</sub> , T <sub>1</sub> , COND, ADDRESS

F<sub>i</sub>: source T<sub>i</sub>: destination

表2 MPUアセンブラ命令

### 3 モニタプログラム

#### 3.1 設計方針

タイプ-1ではマルチプロセッサ部のランプレートプログラムとCPU部のMPU処理ルーチンの2種類のプログラムが実行される。とこ3で一般に画像処理は、いくつかの基本的な処理の組み合わせによって行われ、処理プログラムの構造を動的に変更しなくても、演算パラメータの変更等で済む場合が多い。そこで、タイプ-1での画像アプリケーション処理を、マルチプロセッサ部における画像基本処理に対する基本ランプレートプログラムの順次実行と、CPU部におけるMPU処理ルーチンとの組み合わせ実行、により行うことにした。また、モニタプログラムからは各ランプレートプログラムとMPU処理ルーチンとを共に基本処理単位のプログラムモジュールとして共通に扱うことにした。

プログラムモジュール間のリンク構造は、スタティックなプログラム構造であることを考慮し、構造が簡単で制御の容易な二進木構造で取扱うことにした。しかし、二進木構造であっても、プログラムモジュールに対するif~then~else~等の動的なプログラム構造の変化に対応してゆく余地はある。

#### 3.2 仕様

以下、機能面からモニタプログラムの仕様について述べる。

(1) ホストミニコンとのやりとり

ホストミニコンとマルチプロセッサ部のImage Memory 間の画像ファイル転送と、画像アプリケーション処理コマンドの実行を制御する。画像ファイル

転送の際に必要な、画像のサイズ、転送サイフレームサイズ、データのバイト長さの指定は、画像属性情報テーブルを設けておきインデックス指定によって行なう。

(2) プログラムモジュールの実行.

a. 基本テンプレートプログラム

初期セリトークンの送出と、パラメータトークンと起動トークンの送出との間の overhead を削減するため、プログラムモジュールでは、First Token Stream (FTS) と、Second Token Stream (STS) の二つのトークン群の送出を可能とした。FTS, STS は CPU 部テーブルメモリに格納されており、先頭アドレスとトークン数を指定する。処理の終了は、終了トークン数のカウントとタイマカウントとのどちらかが選択できる。

b. MPU 処理ルーチン

プログラムの指定は program number により行なう。また画像属性情報のインデックスと3つの共用パラメータを与えることができる。処理の終了はモニタプログラムへのリターン命令の実行による。

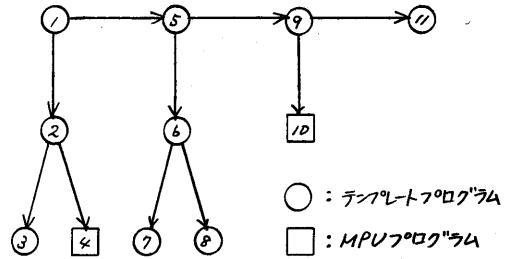
(3) プログラムモジュールのリンクage

二進木構造にリンクする。(図4) 下位側へのリンクをそれぞれを1, 2リンクと呼び、2リンクの終了後は上位プログラムモジュールへ戻す。各プログラムモジュールにはインデックスが割り当てられており、リンクはインデックスで指定する。

(4) ループ処理

テンプレートプログラムモジュールでは、STS の送出に対してループ処理が可能である。さらに二進木の終端でなければ、次の二つのループモードがある。(図5)

i) FTS の終了後、STS を指定数だけ繰り返し処理し、終了後1リンクの実行に移るモード。



番号は実行順序を示す  
処理の開始は任意のノードを指定できる

図4. プログラムの二進木構造

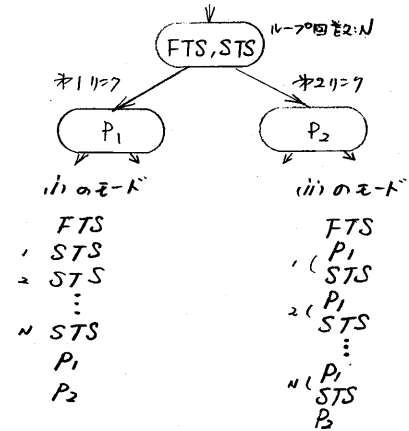


図5 ループ実行制御のモード

テンプレートリスト	FTS 先頭アドレス		
	" トークン数		
	STS 先頭アドレス		
	" トークン数		
	終了条件	リンク数 or タイマカウント	
	ループ回数		
MPU リスト	1リンクインデックス	MPUプログラム指定	ループモード
	2リンクインデックス	"	"
	プログラム・ナンバー		
	画像属性情報インデックス		
	パラメータ 1		
	" 2		
" 3			
ループ回数			
1リンクインデックス	MPUプログラム指定		
2リンクインデックス	"		

図6 プログラムモジュールリストテーブル

(ii) FTS の終了後、第1リンクプログラムモジュールの実行に切り第1リンクの実行とSTSの実行とを繰り返すモード。

ループ処理のネスティングは、別のモードを利用する。ただし、ループ処理は第1リンク側だけであり、ループ終了後は第2リンク側に進む。

MPU処理ルーチンでもループ処理指定ができる。この場合はMPU処理ルーチンと第1リンクとの繰り返しが行なわれる。

各プログラムモジュールの制御情報は、8 word (16 bit/word) のプログラムリンクエスタブルとなっている。(例6)

### 3.3 MPU処理ルーチン

汎用的に利用できるいくつかのプログラムを用意した。これらはパッケージルーチンとして、ユーザーの画像アプリケーション処理プログラム作成の負担を大いに軽減する。用意したMPU処理ルーチンを表3に示す。ただし、マルチプロセッサ部との入出力を伴うものに関しては、その部分のテンプレートプログラムをユーザーが作成する必要がある。

## 4. 画像処理実験

### 4.1 リアルタイム処理

ジョイスティックから入力したパラメータ値を用いて処理した結果を、リアルタイムでモニタディスプレイに表示する。処理として

- (1) 3x3 フィルタ
- (2) 座標変換 (球面鏡歪)

を行ってみた。モニタディスプレイは256x256 (モノクロ) である。使用したMPU処理ルーチンは、「ジョイスティック」による座標値の読み出しと、「トークン修正」による座標値から演算パラメータトークンへの変更である。これらとテンプレートプログラムとをループさせたところ、十分にジョイスティックの動きに追従したリアルタイム処理が行えることがわかった。

### 4.2 FFT

テンプレートプログラムは図2の1次元1ステージ分のバターフライ処理を用いた。「トークン修正」のMPU処理ルーチンによって、ステージ毎に、Weight読み出しのアドレスに対するマスクを更新しながらループ処理で実行する。2次元FFTの場合、ピン

プログラム名	機能
画像転送 (入力)	ホストミニコンとの間で、画像属性情報をもとにデータ転送を行ない、マルチプロセッサ部のイメージメモリに着入る。
画像転送 (出力)	イメージメモリから読み出したデータをホストミニコン側に転送する。
最大・最小値検出	イメージメモリから順次読み出されるアドレスの最大値・最小値及び座標値を検出する。
浮動小数除算	除数、被除数、商の各アドレスをパラメータで指定する。
トークン入出力	マルチプロセッサ部へトークンを出力したり、入力トークンをCPU部イメージメモリに順にストアする。
トークン修正	CPU部イメージメモリ内のトークンデータ増分を加算したり、ビット単位でマスクする。
ジョイスティック	モニタディスプレイ上のカーソルの座標と対応するデータ値を読み取る。

表3 MPU処理ルーチン

トリバースとコーナーターニングを行ない、再び2次元FFTを実行するようループを構成すればよい。ただし、速度的には2次元のアルゴリズムに変更した方が有利である。

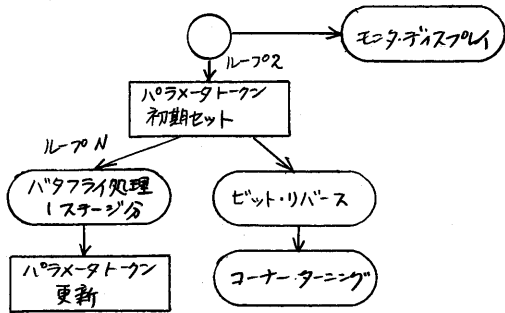


図7 2次元FFTのTree構造

#### 4.3 複合処理

複合処理として簡単なパターンマッチングの実験を試みた。処理の流れを図8に示す。MPU処理ルーチンとして、「画像転送(入力)」、「最大・最小値検出」、「移動小数除算」、「トークン修正」を使用しており、Treeのノード数は30程度となる。

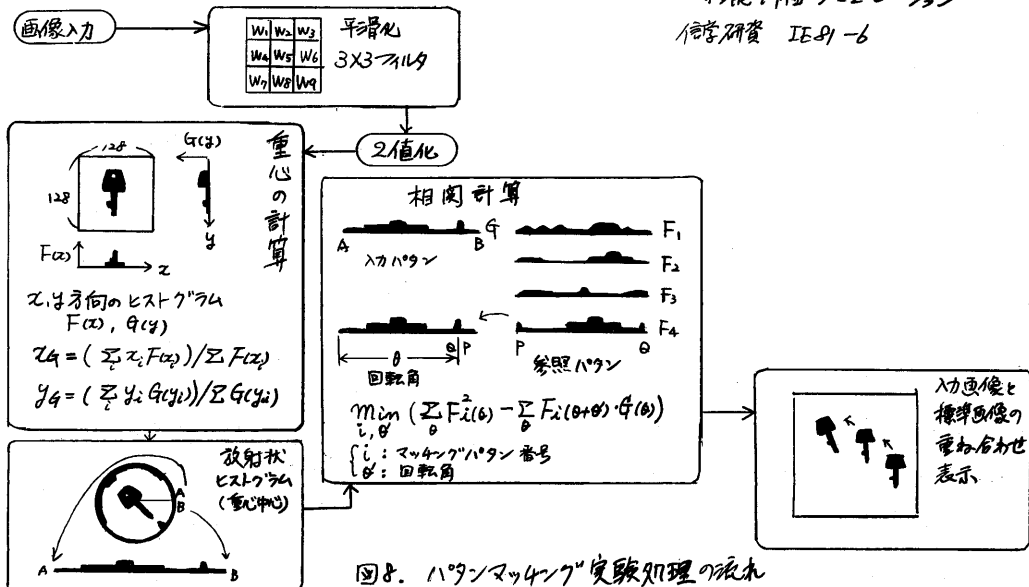


図8. パターンマッチング実験処理の流れ

#### 5 あとがき

TIP-1のマルチプロセッサ部でのテンプレートプログラムとCPU部のMPU処理ルーチンとを実行制御するモニタプログラムの開発を行ない、画像処理実験による評価で良好な結果を得た。今後はMPU処理ルーチンの充実を計り、機能拡張について検討したいと考えている。

最後に、日頃御指導頂く周辺機器研究部花本部長に感謝すると共に、テンプレートアセンブラの作成を担当して下さいました金子様、MPUアセンブラ及びモニタプログラムの作成を担当して下さいました林浩一郎氏に深謝します。

(参考文献)

天橋他, "データ駆動型プロセッサ TIP-1での画像処理プログラミング"  
信学研査 ALP-83 PRL 81-62

天橋他, "画像処理プロセッサ TIP-1の構成と性能評価シミュレーション"  
信学研査 IE81-6