

カラーグラフィックス制御システム G-PSYCO

COLOUR GRAPHICS CONTROL SYSTEM G-PSYCO

久保 正 敏

MASATOSHI KUBO

阿 草 清 滋

KIYOSHI AGUSA

大 野 豊

YUTAKA OHNO

京 都 大 学 工 学 部

FACULTY OF ENGINEERING, KYOTO UNIVERSITY

[1] はじめに

コンピュータグラフィックスは高度なマンマシンインタラクションが要求される分野で利用されている。図形表示のためには、多くの演算が必要であるため、並列化による高速処理能力を持つカラーグラフィックス制御装置にこの演算を任せ、ホスト計算機の負担を軽減するとともに高速処理することが望まれる。我々は、カラーグラフィックスの高度な利用を意図しており、面々色付け、陰影の処理等、大量データの高速処理のために、並列マルチマイクロプロセッサシステム (PSYCO - Parallel processors with SYNchronized Control system) を開発中である。このシステムでは3次元カラーグラフィックスを対象とし、PSYCOを含むグラフィックス制御システムをG-PSYCO (Graphics with PSYCO) と呼ぶ。

G-PSYCOは、人間の視覚が重要な役割を果たす分野への応用に耐え得る高品質の画像と、会話性の向上のため高速処理、という2つの相互する目的を持つ。

G-PSYCOでは、高品質の3次元カラー画像を得るため、画面各点の輝度算出アルゴリズムを従来と異なるものも採用する。従来の3次元グラフィックスでは、対象となる物体を凸多面体で近似し、輝度算出を面単位で行なっているものが多い。また、陰影処理も、凸多面体であることを利用したり、視点と光源の位置を一致させて陰影の問題を避けたものも多い。

我々は、対象とする3次元物体に制限を設け

ず、複数の点光源を視点に対して任意の位置に置くことを許す。また、画面各点の輝度算出は点単位に行ない、同一面内の点であっても、光源との位置関係によって色彩を変化させ、高品質のカラー画像を得るものとする。このためには、膨大なデータ処理が必要となるが、我々は複数のプロセッサの並列構成を考慮し、高速化をはかる。

本稿では、カラーグラフィックス制御装置G-PSYCOの構成とその方針、マルチマイクロプロセッサシステムPSYCOの制御方式について述べる。

[2] グラフィックス処理の流れとシステム構成

一般に3次元グラフィックスシステムで必要な処理は、次の過程と考えられる。

1. 3次元立体モデルの創成

ここで言うモデルは、現実の立体はもちろん、抽象的な概念を立体モデルで表わす場合も含められる。

2. モデルの変形

問題解決の環境に従い、立体モデルを移動、結合、相貫、変形させたりする。

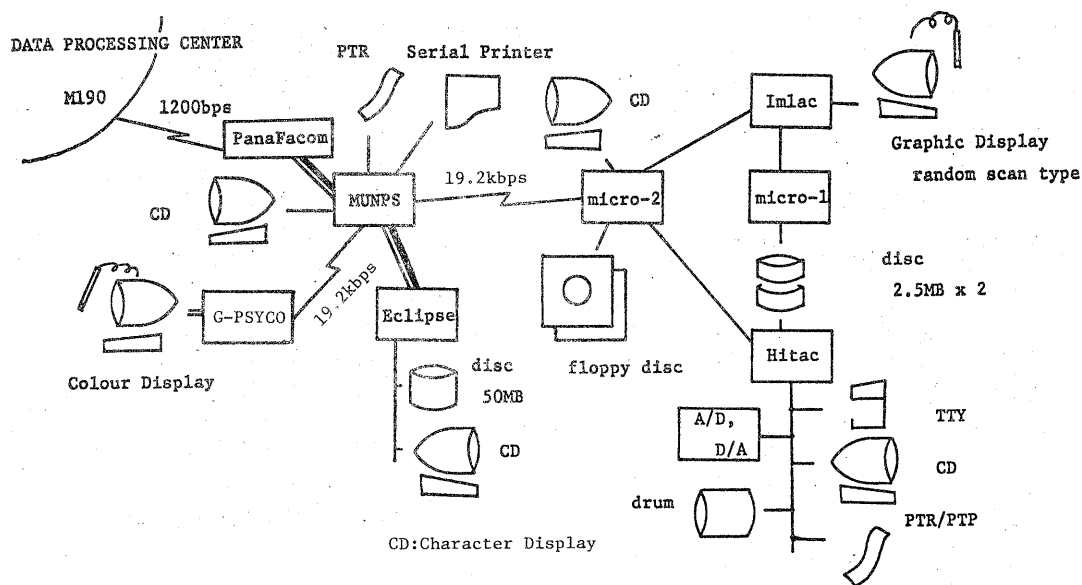
3. 表示画像の生成

生成された立体モデルを人間の視覚に訴える色彩画像に変換する。通常の立体モデルの表示では、モデル上の各点と、視点、光源の位置関係、各点の材質等をパラメータとして各点の色彩を決定するが、問題によっては、他のパラメータによるカラー表示も考え得る。

3次元立体モデルの各点に、100メータに燃
 えて色付け、陰影処理等を行なう第3の過程は、
 データ量、処理時間ともに最も大きくなり、我々は
 この過程を並列マルチマイクロプロセッサシステム
 を用いて高速に処理することを目指している。

G-PSYCOは、計算機制御装置MUNPS (Microcomputer - Used - Network/Peripheral-control - System)²⁾を介し、計算機網の一員として当研究室のシステムに統合されている。MUNPSに

MUNPS は、回線交換型の計算機制御装置であり、マクロコンピュータ (Z-80, 4KバイトRAM, 0.25KバイトROM) によって制御されている。計算機側のOSに大きな変更なく、細を介して各種の資源にアクセスできるようにするために、各プロセス間通信路に、仮想的なDMAインタフェース回路が割付けられる方式が採用されている。これは、プロセスからの回線要求毎に1つのDMAハンドラが生成され、それには互いに独立な機器番号が与えられる。MUNPSはプロセス間通信路の他端のプロセスのデータ転送要求に従い、DMAでデータを転送し、指定の動作終了時には、与えられた機器番号を持って割り込み処理を要求する。これにより、計算機側に要求されるソフト



— 68 —

ウェアは、プロセス間通信の起動と終了割込み特許のみとなる。MUNPSには、PTR、プリンタ、キャラクタディスプレイなどの入出力機器が接続されている。仮想インタフェース回路により、物理的には1つのDMAインタフェースをバイト毎に複数の通信路が多重化される。PTRなどの入出力装置にはバッファリング機能がなく、データブロック毎の多重化を行なうとすれば、物理インタフェースの利用効率是非常に悪いものとなる。プロセス間通信のための機能は極めて限られたものにして、低速なマイクロコンピュータで制御可能なものとしている。例えば、プロセス間通信路設定の同期の簡略化、エラー対策の簡略化などが図られている。

[4] G-PSYCOのハードウェア構成

G-PSYCOのハードウェア構成を図-2に示す。ホスト計算機から与える立体モデルを表示画像に変換する機能を持つので、ホスト計算機からは、ディスプレイプロセッサとして見なすことができる。

G-PSYCOは、ディスプレイプロセッサとして必要な機能をモジュールに分解し、それぞれをファームウェアで実現した構成となっている。即ち、ディスプレイコントローラはG-PSYCOの核となるモジュールあるいはタスクスケジューラ、並列マルチマイクロプロセッサシステムPSYCOは演算装置、I/OプロセッサはI/Oハンドラをそれぞれファームウェア化したものと考えられる。

4.1 I/O プロセッサ

I/Oプロセッサはマイクロコンピュータ(8080, 4KバイトRAM, 1KバイトROM)で構成され、タブレット、キーボード、ライトペン等が接続されている。I/Oプロセッサは

- ・G-PSYCOを当研究室の計算機網に参加させるためにMUNPSとインタフェース条件を整合させ、ホスト計算機とG-PSYCOとの通信制御を行なう。

- ・複数周辺機器のI/O処理の管理

といった機能を集約して行なう。このプロセッサのメモリ空間の一部はディスプレイコントローラメモリ空間と共有されているので、ディスプレイコントローラから見れば、マルチプレクサチャネルとしての性格も持っている。

4.2 ディスプレイコントローラ

ディスプレイコントローラは、ホスト計算機から与えられる立体モデル、周辺機器から与えられるユーザコマンドに従い、必要な画像演算を演算装置を用いて実行し、演算結果を編集してディスプレイファイルは画像メモリ上に作成する。我々はこの画像演算装置を並列マルチマイクロプロセッサで構成し、PSYCOと称している。ホスト計算機で得られた立体モデルを受取って表示画像に変換するために必要な演算は、マトリクス演算が殆んどであり、大量のデータに対して繰返し同一の演算を行なうことが多い。

ディスプレイコントローラはディスプレイ

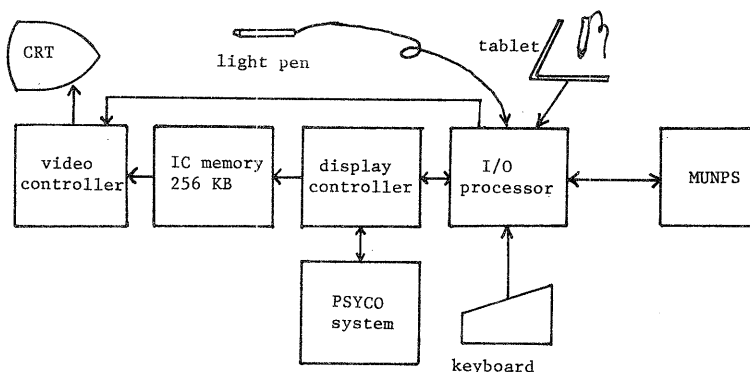


図-2 G-PSYCOのハードウェア構成

ファイルを作成するためにPSYCOを起動する。この時、PSYCOの各プロセサには、プロセサ間の通信が不要であり互いに独立に異なるデータに対して同一の演算処理を行なわせるとすれば、PSYCOの構成が簡単になるのは明らかである。このため、ディスプレイコントローラは、必要な画像演算を互いに独立で並列処理可能な演算に分解し、PSYCOの各プロセサに分配して演算を開始させる。ここで注意する点は、PSYCO側では、例えば自分が行なう輝度算出演算が表示画面上のどの点のものであるかと言ったグローバルな知識を与えられておらず、単に複数の演算器としての役割を果たす点である。演算器の機能に徹することにより、マルチマイクロプロセサの構成が簡単化される。演算結果をどう関連づけるかと言ったグローバルな知識はディスプレイコントローラが持っており、PSYCOが生成した演算結果のデータを収集、編集しながらディスプレイファイルを作成し、画像メモリに書込む。

すなわち、ディスプレイコントローラは、画像演算に必要なタスクを並列処理可能な子タスクに分解してPSYCOに分配し、それらの終了に従って新たに別のタスクを分配して起動するというタスクスケジューラの機能、子タスクが生成したデータを編集するデータマネージメントの機能を持つ。

ディスプレイコントローラは、演算装置としてのPSYCOとのデータの授受、画像メモリへのデータの書込みを行なうので、データ転送に

よるオーバーヘッドを小さくするために、画像メモリ、及びPSYCOと密に結合されている必要がある。我々はこの観点から、ディスプレイコントローラが、PSYCOのメモリ及び画像メモリを自分のメモリ空間として直接アクセスできる構成をとっている。言い換えると、ディスプレイコントローラは、PSYCOの並列プロセサ群とそれらのメモリを共有し、またビデオコントローラと画像メモリを共有している。また前述の様に、I/Oプロセサともメモリを共有している。これらの関係を図-3に示す。

この点で、ディスプレイコントローラは、メモリマネージメントプロセサ、あるいは周辺に結合されたプロセサ群から見ればDMAコントローラとしての性格も持っている。

ディスプレイコントローラはこのように大きなメモリ空間を持つ必要があるため、我々は、1Mバイトのアドレッシング能力のあるIntel-8086を用いて制御する。

ディスプレイコントローラは、上述の如く高速の演算装置を持つ強力なプロセサとしても捉えることができ、将来は[2]章で述べた1, 2の処理も担当させ、ホスト計算機に準にファイルマネージメントのみを行なわせる予定である。

4.3 ビデオコントローラ

画像メモリを順次読出し、D/A変換してCRTへ送出し、画像を作成する部分である。CRTは2Dインテリジ解像度カラーCRTを用

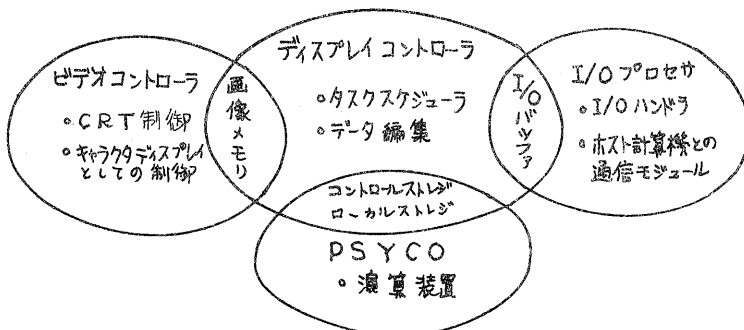


図-3 G-PSYCOの機能分担

い、 512×512 の解像度を持つ。画像メモリには256 KバイトのICメモリを用い、1画像データはR, G, B各々3ビット計9ビットである。

画像メモリ読出しのための順次アドレスの生成、同期信号の生成等はCRTコントロールチップを用いて集積化を図っている。

グラフィック表示に文字を重畳して表示することが可能となるように、画像メモリとは別に2Kバイトのキャラクタ格納用RAMとキャラクタジェネレータを置き、この制御のためにマイクロコンピュータ(ヌ-80, 128バイトRAM, 1KバイトROM)を用いている。

[5] PSYCOの制御方式

プロセシングエレメント(以後PEと呼ぶ)を用いた並列プロセサは従来提案されているが、我々が狙いとするのは、複数データストリームに対して同一の演算を並列に処理するSIMD方式の並列プロセサである。

複数のPEに対する演算命令系列すなわちプログラムは同一であるから、1つずつ有メモリ(コントロールストレジ 以後CSと呼ぶ)に格納すればよく、各PEに与えられるオペランドとしてのデータは、それぞれにローカルメモリ(ローカルストレジ 以後LSと呼ぶ)に

格納される。CSに格納されているプログラムは同一であるが、各PEが実際にプログラムを実行する時に観測される命令実行系列はデータの値に従ってそれぞれのPE毎に異なるのが一般であり、各PEが命令語をフェッチするために行なうCSへのアクセスはランダムと成って競合を生じる。この競合を回避するためには、CSと各PEの間でバスアービタが必要となる。この様な構成を図-4(a)に示し、非同同期型制御方式(A型制御方式)と呼ぶことにする。アービタを使用しても、並列化させるPEの台数には限界があると思われる。

これに対し、プログラムの命令実行系列がデータに依存せず、任意の時点で各PEが実行する命令語がすべてのPEについて同一であれば、各PEのCSに対するアクセス競合は考慮する必要がなく、並列化するPEの台数にも制限がないと考えられる。この場合の構成をステップ同期型(S型)制御方式と呼び、図4-(b)に示す。

しかし、各PEの命令実行系列が等しくなるようなプログラムは一般的でなく、従来のアルゴリズムを変換する必要もある。また、変換に伴うプログラム長の増加、実行時間の増加などの程度になるか問題となる。

5.1 ステップ同期型プログラム

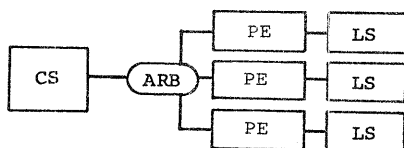
S型の制御方式では、ステップ同期型プログラム(S型プログラム)も実行することになる。

定義: ステップ同期型プログラムとは、実行開始後ある時点で実行される命令語は、データに依存せず同一であり、このことかどの時点についても言えるようなプログラムである。即ち、データに独立な制御フローを持ち、かつ、ある命令語の実行時間もデータに依存しないようなプログラムである。

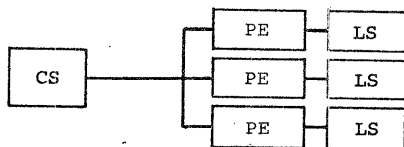
何を用いてPEを実現するかによって、様々なレベルの並列プロセサシステムが考えられ、それに対応したレベルのS型プログラムを開発する必要がある。PSYCOではPEとして、マイクロコンピュータを採用する。その理由は次の点である。

- 並列化されたPEの機能が大きいこと。

即ち、並列プロセサシステム自体でまとまったタスクを実行でき、外界との通信が



(a) A型…非同同期型制御方式



(b) S型…ステップ同期型制御方式

PE: Processing Element
CS: Control Storage
LS: Local Storage
ARB: Bus Arbiter

図-4 並列プロセサ制御方式

減少する。

- ・並列プロセカシステムの実現が容易。

即ち、実装の容易さ、コストの小ささ。

我々は、マイクロコンピュータの命令語のレベルでのS型プログラムを開発する必要がある。

通常の四則演算をS型プログラムに変換することは、プログラムフロー上の、データに依存した分岐を表面上なくすることである。変換のための手法の骨子は

- ・forkからjoinまでのそれぞれのパスで得なければならない処理結果のうち、それを生成するためには、プログラムレベルで異なる処理（PCとIRが異なる得る処理）を必要とするものに対しては、それらの処理をすべて行ない、結果を用意する。
- ・joinで、制御変数に従って、そのうちの1つの結果を選択する。選択操作は、プログラムレベルで全く同一の操作であること。

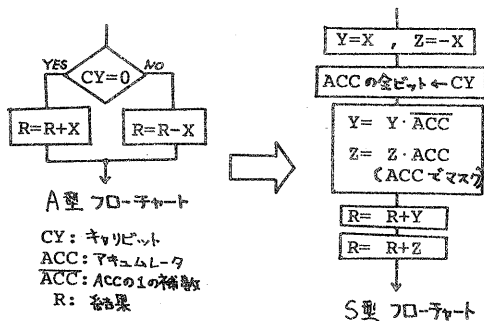


図-5(a) S型への変換法 1
(マスキング)

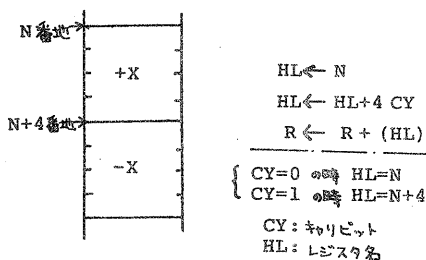


図-5(b) S型への変換法 2
(アドレスレジスタの操作)

具体的な例として、キャリビットの値によって今後加算なり、中間結果RにXを加える場合と減ずる場合についての2つの変換方法を図-5(a),(b)に示す。(a)の方法は、キャリビットを使ってマスクを作製し、各分岐先の演算結果をすべて用意し、各結果にマスクをかけるか否か重ねることにより必要の結果のみを選択する方法である。(b)の方法は、各分岐先の演算結果をすべて求めてメモリ上に並べ、アドレスレジスタを制御変数に従って操作することにより、どの結果を選択するかを決定する。(b)の方が、処理速度が大きい。

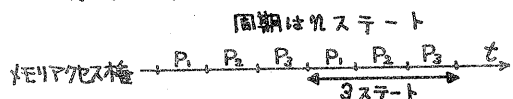
5.2 制御方式の比較

我々はA型とS型の制御方式を比較するため、PEとしてIntel-8080を仮定して、処理時間の予測を行なった。8080において、

1. 1命令は1~5個のマシサイクルに分割される。
2. 1マシサイクルは3~5個のマシステートに分割される。
3. 各マシサイクルは命令語のフェッチ、データのI/O等メモリに関する処理の単位である。
4. メモリとのI/Oはそのマシサイクルの3ステート目まで実行される。
5. 命令語は1~3バイト長である。
6. ステートはマイクロプロセカの最小のタイミング単位でクロック ϕ_1 , ϕ_2 に同期している。
7. T_w というステートはメモリのアクセス時間の調整に使用されるものでREADY信号で制御する。

S型と比較するためのA型の制御方式は、図-4(a)に示すものである。バスのアービトレーションは、ステート単位に n 台のPEにCSに対するメモリアクセス権を与えることにより実現する。すなわち、各PEがメモリアクセスを生じた時、アクセス権があれば T_3 、なければ T_w となる。

$n=3$ の時について示す。



5.2.1 処理時間の見積り

次に示す2種類のデータ形式を考えた。

- { F型 …… 浮動小数点4バイトデータ形式
- { I型 …… 固定小数点4バイトデータ形式

このそれぞれについて四則演算と開平演算の5つのルーチンを作製した。なおプログラムはA型、S型を考えたのでFA, FS, IA, ISと4種類である。乗算はRobertsonの方法、除算と開平は引き放し法を採用した。各々のルーチンについて実行ステート数を見積った。A型は各分岐確率を等しいと仮定した。見積り結果は、下位のルーチンも含めたプログラム長と共に表-1に示す。

加減算の処理ではF型の場合にデータの正規化が必要で処理時間がかかる。乗除算では正規化の処理の影響は少ないが、有効数字の桁数が多いI型では処理時間がかかる。開平は共に処理時間が大きい。

	演算名	S型	A型
F型	加算	40	6
	減算	40	6
	乗算	15	9
	除算	21	18
	開平	142	130
I型	加算	0.3	0.3
	減算	0.6	0.6
	乗算	19	11
	除算	31	28
	開平	300	284

	S型	A型
F型	845	730
I型	599	585

プログラム長
(バイト)

実行ステート数
(×1000ステート)

表-1 実行ステート数とプログラム長

5.2.2 処理時間のシミュレーション

A型の制御方式の各PEに与えられる命令語を追跡し、アービタの動きと各PEのプログラムカウンタの進捗を模擬するシミュレーションプログラムを作成した。5.2.1で作成した各プログラムをデータとしてA型制御方式の処理時間をシミュレートし、S型の処理時間と比較したものの一部を図-6に示す。

加算の場合、S型の性能はA型より著しく劣っている。これは下型データに対する正規化の処理が影響したためである。しかし、加減算以外では、教台以上の並列化を行なうならば、A型より優れているといえよう。処理の種類により、演算の種類と頻度を考慮し、また並列化台数の大きい処理の場合にS型の制御方式を用いるならば、構成上の利点、制御方式の簡易性などを発揮しうる。

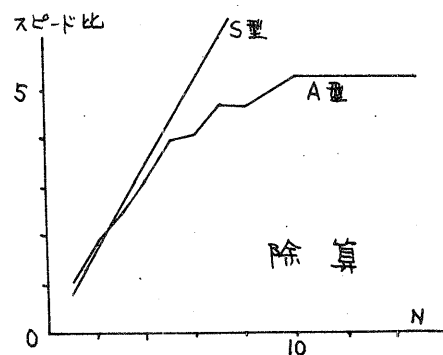
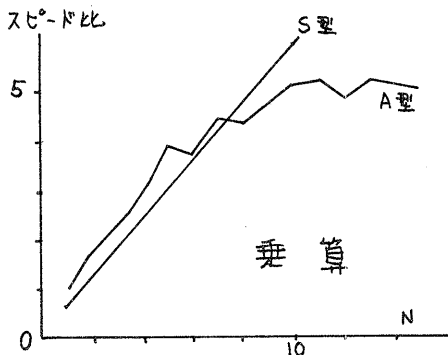
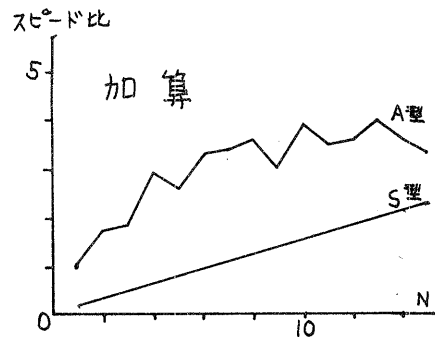


図-6 F型データによる処理速度比較

5.2.3 複合計算における処理時間の比較

各演算ルーチンの中では、各種シフトルーチンが実行時間に占める割合が非常に大きい。そこで、シフトルーチンをハードウェアで実現することにより、処理時間の高速化を図る。

LS 部をシフトレジスタで構成し、左右 1 ビットシフト、n ビットシフト、係数部の正規化などの機能を 1 命令で実行させる。このようにハードウェアでシフトを持つ場合に、5.2.1 で作成した各ルーチンの実行ステート数を見積り、表-2 に示す。表-1 と比較して、かなりの改善が見られる。

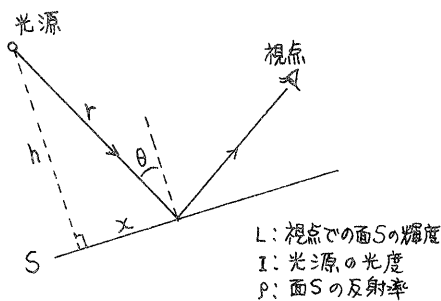
	演算名	型	型
F	加算	1.7	0.4
	減算	1.9	0.6
	乗算	9.4	4.6
	除算	14	11
	開平	39	27
I	加算	0.3	0.3
	減算	0.6	0.6
	乗算	16	7
	除算	21	18
	開平	66	49

	型	型
F 型	677	574
I 型	494	471

プログラム長
(バイト)

実行ステート数
(×1000 ステート)

表-2 実行ステート数とプログラム長
(シフト付)



$$L = \frac{\cos \theta}{r^2} \cdot I \cdot \rho$$

$$L(x) = \frac{h}{(h^2 + x^2) \sqrt{h^2 + x^2}} \cdot K$$

図-7 輝度算出演算

このような構成で PSYCO を実現した時に具体的に行なうべき演算について処理速度を見積る。PSYCO で行なう処理の中で最も処理時間を必要とするのは、図-7 に示す輝度算出演算である。この演算の実行速度を表-2 の各ルーチンの処理速度比に使用回数、実行時間の重みを加えて F 型、I 型に対し求めた結果を図-8 に示す。他の計算機の処理能力との比較のために、ハードウェアで 16 ビットの固定小数点データの乗除算ルーチンを持つミニコンによる処理時間を見積った。浮動小数点 4 バイトデータにつき、約 9.2 ms 要する。これは、S 型の制御方式で下型データを採用した場合、約 5 台並列化した処理能力に対応する。

[6] PSYCO の設計仕様

前章の検討結果を基に PSYCO の設計仕様を決定する。このシステムの目的は、高品質のカラー画像を高速に行なうことである。このために最も重要なのは輝度算出演算であり、また画像生成に必要時間の大部分を占めるのも、

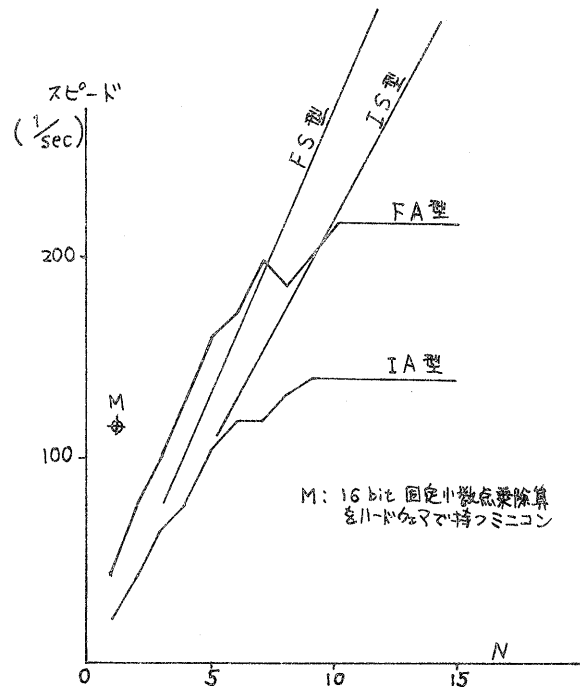


図-8 輝度算出の処理速度比較

この演算である。従って PSYCO の設計仕様を決定するに際し、輝度算出に必要な処理時間の予測値をもとに制御方式を決定するのが妥当である。我々は、シフト付きのステップ同期型制御方式を以下に述べる理由により採用する。

1. 図-8 により、非同期型制御方式による処理速度は、プロセッサ台数がほぼ 10 台で頭打ちであり、それ以上の速度向上は望めない。
2. ステップ同期型制御方式では、処理速度はプロセッサ台数に比例するので、台数を増せば更に高速化が図れる。
3. ステップ同期型制御方式では、バスアービタは不要で、簡単なデータバススイッチがあればよく、ハードウェアの構成が単純である。従って、製作途中でのチェックが簡単で保守性がよい。
4. 3 の理由により、ステップ同期型制御方式のマルチマイクロプロセッサシステムでは、プロセッサの増設が簡単で、既存の部分の変更が少なく、従ってシステムの拡張性がよい。

PSYCO の構成図を図-9 に示す。ディスプレイコントローラから PSYCO の CS, LS へのアクセスは、すべての PE をホールド状態にして行なう。図からも、システムの拡張性のよいことがわかる。

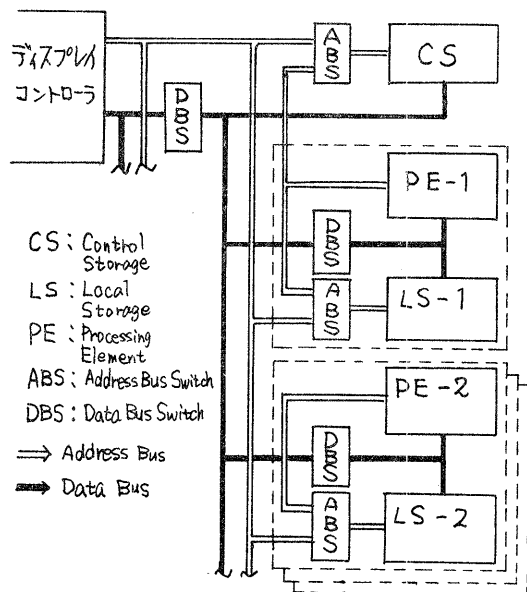


図-9 PSYCOの構成図

プロセッサの台数 N を 16 台とし、PSYCO による色彩決定処理時間を予測してみる。

[5] 章の処理時間見積りの結果より、F 型データを用いて画素 1 点分の R, G, B 成分算出に要する時間が次式で求まることになっている。

$$T = \frac{130}{N} \times 1000 \text{ ステート}$$

1 ステートを $0.5 \mu\text{s}$ とすると画面全体の輝度算出には、 $N = 16$ として

$$\frac{130}{16} \times 10^3 \times 0.5 \times 512 \times 512 \mu\text{s} = 17.75 \text{ min}$$

必要となる。しかし、実際には全画面の画素について輝度算出することはなく、また輝度が滑らかに変化する部分については内挿を行ない、演算を簡単化できる。これらを考慮すれば、約 10 分の 1 に減少すると考えれば、約 2 分で 1 画面が得られ、マンマシン対話性からみても、充分高速であると言えよう。

[7] おわりに

G-PSYCO は、高品質のカラー画像を高速に得るためのディスプレイ制御装置で、そのために必要な機能を分解してファームウェア化し、演算機能を高速化するために、独特の制御方式を用いた並列プロセッサシステム PSYCO を包含する。この過程で考案したステップ同期型プログラムは、いろいろなプログラムレベルで考えられ、プログラム検証等への応用についても考察中である。

現在、ハードウェアの詳細設計をほぼ完了し、製作途中である。

文献

- (1) K. Agusa, Y. Ohno: "MUNPS: Microcomputer Used Network/Peripheral Control System" 12th HICSS 1979 発表予定
- (2) 亀井, 久保, 阿草, 大野: "カラーグラフィックス向き演算用マルチマイクロプロセッサシステム" 通信学会研究資料 EC77-57
- (3) W. Jack Bouknight: "A Procedure for Generation of Three-dimensional Half-toned Computer Graphics Presentations", Comm. ACM, 13, 9 (1970, 9)
- (4) D. F. Rogers: "Mathematical Elements for Computer Graphics", McGraw-Hill (1976)