3次元色彩画像表示処理専用マルチマイクロプロセッサシステム G-PSYCO†

久保正敏†† 戸島英一朗††† 亀井朗†††† 阿草清滋††††† 大野豊†††††

われわれは，マイクロプロセッサ複合体によって3次元カラーグラフィックスシステムの画像表示処理専用プロセッサを開発し，G-PSYCO と名づけた．G-PSYCO は，画像表示処理に含まれる SIMD 型処理の高速化をねらった並列プロセッサシステム PSYCO を含む．PSYCO は，DIF プログラム概念を用いたステップ同期制御方式と呼ぶ独特の方式で制御されている．DIF プログラムは，データの値に依存しない制御フローをもつプログラムであり，従来の SIMD マシンではハードウェアで行われてきたデータ依存分岐の実行選択機構をプログラム内で実現したものである．ステップ同期制御方式を用いることにより，拡張性，保守性の優れた非常に簡単なハードウェア構成で SIMD マシンを開発することができた．本論文では，G-PSYCO のハードウェア構成，PSYCO の制御方式，特徴を中心に，3次元色彩画像表示における並列処理手法，簡単な性能評価について述べる．

1. まえがき

コンピュータグラフィックスの分野では，3次元グラフィックスシステムの需要が大きくなっているが，ラスタスキャン型 CRT を用いて3次元色彩画像表示を行うには，座標変換，隠面消去，面の色つけ，付影処理等の膨大な演算が必要となる．

われわれは，マイクロプロセッサ複合体によって3次元カラーグラフィックスシステムの画像生成端末を開発し，G-PSYCO と名づけた．G-PSYCO は，高速演算装置としての並列プロセッサシステムを含んでいる．この並列プロセッサシステムは，前記の各処理が SIMD (Single Instruction stream Multiple Data stream) 型処理[1] を多く含む点に着目し，マイクロプロセッサを用いて SIMD マシンを実現したものである．

従来の SIMD マシンは，並列 PE (Processing Element) が，SI (Single Instruction stream) に相当するプログラムを保持する一つの CU (Control Unit) に制御される形態であるが，SI がデータに依存する分岐を含む場合には各 PE の命令実行を選択制御する必要があり，このためのハードウェア機構が複雑であった．われわれは，この命令選択機構を，ハードウェアではなくプログラムに埋め込むことにより，ハードウェア構成の簡単化を実現した．すなわち，データに依存する分岐をなくしたプログラムを考案し，DIF (Data Independent Flow) プログラムと名づけた．DIF プログラムを SI とすることにより，全 PE が命令実行ごとに同期をとりながら処理を進めることができるので，われわれはこの制御方式をステップ同期制御方式，並列プロセッサシステムを PSYCO (Parallel processors with SYnchronized COntrol) と呼んでいる．

本稿では，G-PSYCO および PSYCO のハードウェア構成を中心に，その制御方式，特徴，性能評価について述べる．

2. G-PSYCO のハードウェア構成

G-PSYCO は，画像生成端末として必要な機能を分解し，それぞれをマイクロプロセッサで実現したもので，**図1**に示すように，コントロールプロセッサを中心に，ビデオコントローラ，I/O プロセッサおよび並列プロセッサシステム PSYCO が，メモリ共有で強結合された構成となっている．

ビデオコントローラは，256 k×9 bit の画像メモリを順次読み出し，D/A 変換後カラー CRT へ送出して

† G-PSYCO; A Multi-Microprocessor System for Generating Three-Dimensional Color Images by MASATOSHI KUBO (National Museum of Ethnology), EIICHIRO TOSHIMA (Canon Inc.), AKIRA KAMEI (Toyo Engineering Corporation), KIYOSHI AGUSA and YUTAKA OHNO (Department of Information Science, Faculty of Engineering, Kyoto University),
†† 国立民族学博物館
††† キヤノン(株)
†††† 東洋エンジニアリング(株)
††††† 京都大学工学部情報工学科

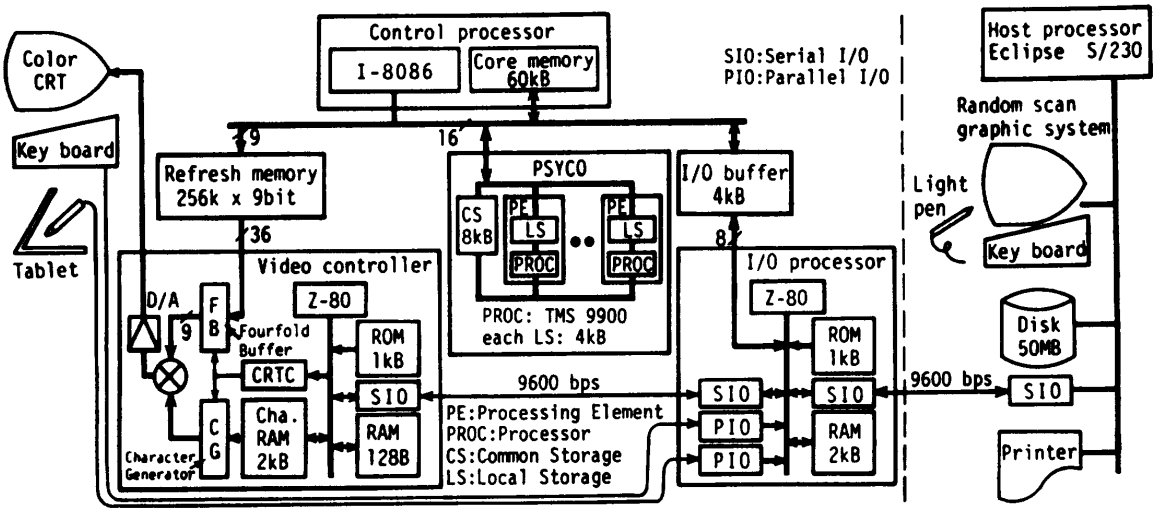


図 1 G-PSYCO のハードウェア構成
Fig. 1 Hardware configuration of G-PSYCO.

画像を表示する．1画素に R, G, B 各3bit を当て，512×512 の解像度を得ている．メモリ読出し速度の不足を補うため，画像メモリを2バンクに分けて 2-way インタリーブを行い，各バンクからは一度に4画素分読み出している．グラフィック表示に64字×32行 の文字を重畳表示するキャラクタディスプレイの機能をもち，このための制御をマイクロコンピュータ (Z-80, 1kBROM, 128BRAM) で行っている．

I/O プロセッサには，キーボード，タブレット，文字出力機器としてのビデオコントローラが接続されており，これら I/O 機器の制御を行うほかに，ホストプロセッサと G-PSYCO との通信制御も司る．I/O プロセッサは，4kB の I/O バッファをコントロールプロセッサと共有している．

コントロールプロセッサは，ホストプロセッサから与えられる立体モデル，入力機器から与えられるコマンドに従い，必要な演算を PSYCO を用いて実行し，演算結果を編集して表示画像データを画像メモリ上に作成する．すなわち，コントロールプロセッサは，必要な演算を並列処理可能な子タスクに分解し，PSYCO の各 PE に分配して起動するというタスクスケジューラの機能，各 PE が生成した並列処理結果を編集するデータマネージャの機能を果たす．この際のデータ転送に伴うオーバヘッドを軽減するため，コントロールプロセッサは各構成要素とメモリを共有しており，その関係を図2に示す．

3. 並列プロセッサシステム PSYCO

3.1 PSYCO の制御方式

PSYCO は，3次元画像表示処理向きの SIMD マ

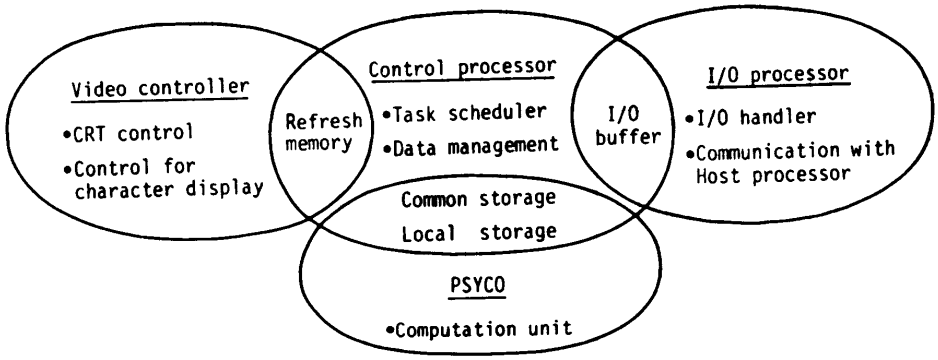


図 2 G-PSYCO の機能分担とメモリ共有関係
Fig. 2 Memory sharing schema and function of G-PSYCO.

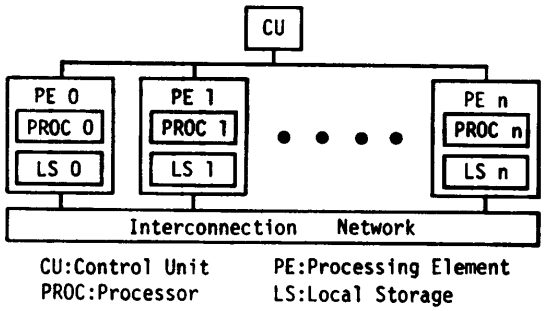


図 3 SIMD マシンのモデル構造
Fig. 3 Model structure of SIMD machine.

シンをマイクロプロセッサで実現したものである．

SIMD マシンには，ILLIAC Ⅳ[3] をはじめとして，DAP[6]，MPP[8]などの著名な開発例があるが，これらはいずれも，図3に示すモデル構造で表すことができる[4]．すなわち，一つの CU (Control Unit) が SI (Single Instruction stream) に相当するプログラムを全 PE(Processing Element) に送出し，各 PE が LS(Local Storage) 内に保持している MD (Multiple Data stream) に対して同時に同一命令を施すものである．一般に SI は，MD の各データに依存する分岐を含み，それを実行する際には，各 PE ごとに異なった処理が必要となる．この問題に対処するため，従来の SIMD マシンにおいては，各 PE がマスクレジスタをもち，CU から送出された命令を実行するか否かを選択したり，CU が二重命令を送出し PE がそのどちらかの命令を選択する[5] 等の方法が採られてきたので，PE や CU のハードウェア構成が複雑であった．さらに，PE 間のデータ授受が必要なアレイプロセッサを指向するあまり，PE 間通信路の制御が煩雑であった．

われわれは，ハードウェア構成の簡単化を図るため，データ依存分岐に対する命令選択機構を，ハードウェアではなくプログラムに埋め込んだ形で実現する手法として，命令語レベルでデータ依存分岐を含まないプログラム，DIF プログラムの概念を考案した．DIF プログラムとして実現された SI を CS(Common Storage) に格納しておけば，各 PE が能動的に命令取出しを行っても，その時刻は互いに同期するため，CS に対するアクセス競合は生じない．したがって，PE の状態を把握しながら命令送出制御を行う CU が不要となり，PE の側でも特別な命令選択機構が不要となる．われわれはこの制御方式を，命令語のステップごとに PE を同期させるという意味で，ステ

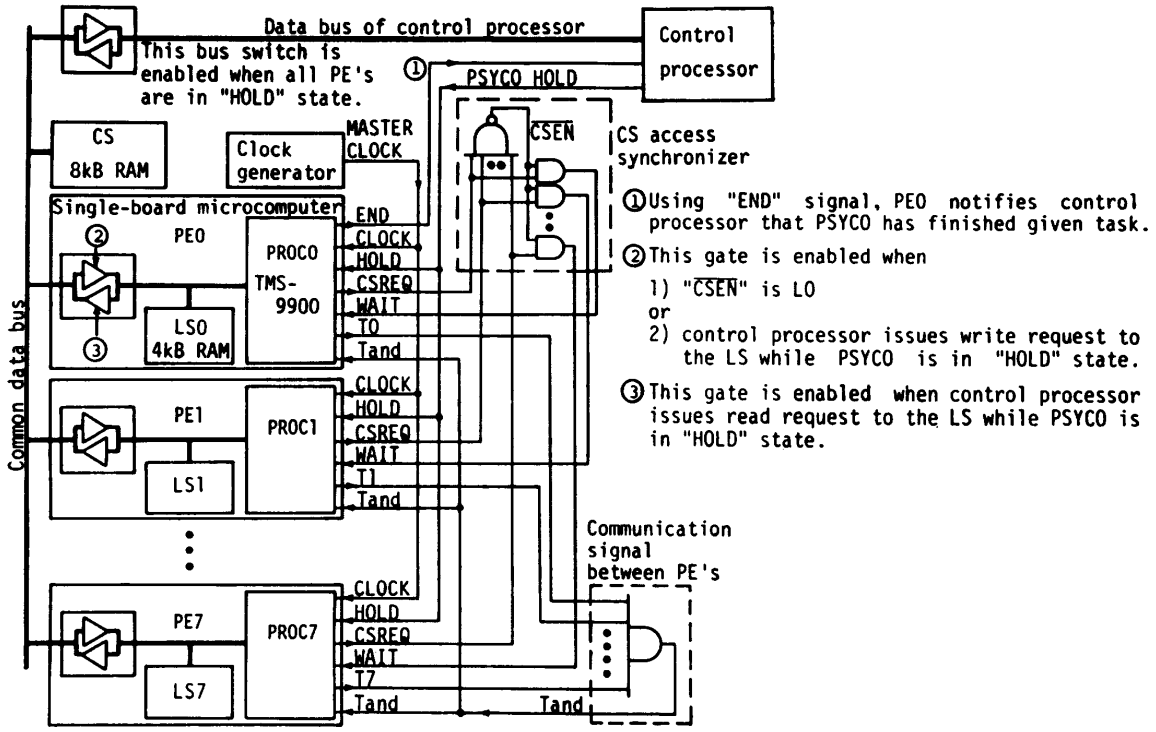


図 4 PSYCO のハードウェア構成概略
Fig. 4 Hardware configuration of PSYCO.

ップ同期制御方式と呼んでいる．

また，3次元画像表示処理に含まれる SIMD 型処理は，後述するように，PE 間で互いに独立な並列処理として実現できるので，PE 間通信路は不要となる．

以上のように，PSYCO は，DIF プログラムの採用によって，簡単なハードウェア構成で SIMD マシンを実現したもので，システムの開発，保守，拡張が容易である．

3.2 PSYCO のハードウェア構成

PSYCO のハードウェア構成概略を**図4**に示す．PE には，TI 社の TMS 9900 を CPU とし，4 kB RAM，PIO 等が実装されたシングルボードマイクロコンピュータを利用した．ステップ同期制御方式を実現するためにハードウェア構成上留意すべき事項として

- 外部のクロックジェネレータで生成した共通クロックにより各 PE を動作させる必要がある，
- データの値によって実行時間の変化する命令語が存在し，DIF プログラムであっても命令語取出し時刻が PE 間で異なる場合が生じる．このため，CS access synchronizer 回路を設け，CS に対するアクセス要求がすべての PE から出揃うまで，要求を発した PE を WAIT させる，
- 命令語取出しのために CS にアドレスを送出するのは PE0 に限定する．PE0 には，PSYCO に与えられたタスクの終了をコントロールプロセッサに通知する END 信号も出力させる，

等があり，これらに対応する改造をコンピュータボードに施している．また，PE 間には，Ti と Tand と名づけた信号線を供給している．各 PEi は Ti を0または1にセットし，全 Ti の論理積 Tand をチェックできる．

PSYCO とコントロールプロセッサとの間のデータ授受は，全 PE を HOLD 状態にして，CS, LS をコントロールプロセッサに解放することにより行う．コントロールプロセッサが全 LS に共通データを送出するブロードキャストモードも備えている．

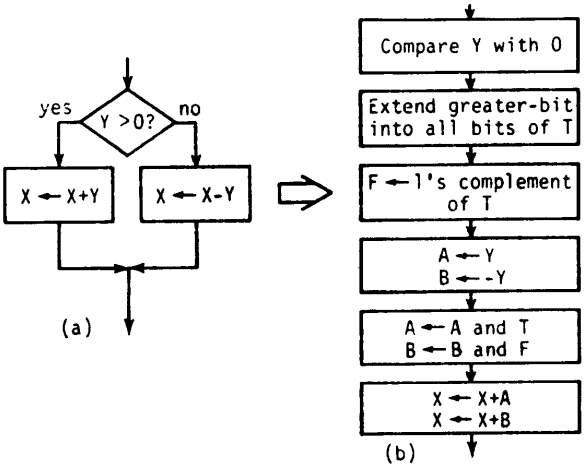


図5 if 構造の DIF プログラムへの変換
Fig. 5 Conversion of if-structure into DIF program.

3.3 DIF プログラム

DIF プログラムとは，データの値に依存しない制御フローをもつプログラムである．通常のプログラムを DIF プログラムへ変換する手法を，if 構造および loop 構造に分けて説明する．

(a) if 構造

- then 部，else 部で得るべき処理結果両者を生成する．
- 2者のうち一つを，condition 部の結果に従い選択する．

Yの絶対値をXに加える**図5**の例で説明する．まずYと0との大小を比較し，対応するステータスフラグをセットする．レジスタTの全ビットに，このフラグビットが拡張されてセットされ，Fにはその1の補数がセットされる．T, F を用いて then 部，else 部に対応する中間結果をそれぞれマスクし，適切な結果を選択する．われわれはこの手法をマスク法，T, F を制御フラグと呼んでいる．制御フラグは，従来の SIMD マシンにおけるマスクレジスタに対応するが，これを用いて命令の実行を選択するのではなく，実行の後その結果を選択するのがわれわれのマスク法である．

(b) loop 構造

不定回数 loop 構造は，次の手法でデータ値に依存しない固定回数 loop 構造に変換できる．

- その loop 構造に固有の最大回数 I_{max} を見いだす．
- loop 中に，必要な処理が終了すれば起動される無効ジョブを else 部にもつ if 構造を形成する．

しかしこの方法では，実効的な loop 回数が I_{max} に比べて小さい場合，非常に効率が悪くなる．DIF プログラムの目的は，並列 PE 間に同一の命令系列を供給することである．したがって，loop に固有の I_{max} を定めるのではなく，PE 群に与えられ

た1組のデータについて最大 loop 回数を定めればよい．このために前述した Ti, Tand 信号を用いる．各 PEi は，自己の loop 終了条件を満たせば Ti を1にセットし，次に Tand をチェックしこれが0であれば無効 loop に入る．loop 内の if 構造は前述の方法で DIF プログラムに変換される．**図6**に，アレイ内の項目を探索する不定回数 loop 構造を例に，DIF プログラムへの変換手法を示す．

3.4 PL/DIF と PL/DIF コンパイラ

DIF プログラムは多くのマスク操作を含むため，そのプログラミングは容易ではない．われわれは，DIF プログラム開発を支援するため，プログラミング言語 PL/DIF とそのコンパイラをホストプロセッサ上に開発し，PSYCO のソフトウェア開発に利用した．

PL/DIF は，次のような制御構造文をもつ他は，TMS 9900 のアセンブリ言語をそのまま用いており，一種の構造化アセンブリ言語である．

(a) IF (cond) THEN (stmt) FI
(b) IF (cond) THEN (stmt 1) ELSE (stmt 2) FI
(c) LOOP (stmt) REPEAT
(d) WHILE (cond) LOOP (stmt) REPEAT
(e) DO (loop-count) LOOP (stmt) REPEAT
(f) WHEN (cond) EXIT

ここで，cond は条件文，stmt は文の系列を示す．PL/DIF コンパイラは(a),(b)の if 構造にはマスク法を，(c),(d)の不定回数 loop 構造には Ti, Tand 信号をそれぞれ用いて DIF プログラムへ変換する．

マスク法には制御フラグが用いられるが，ネストした if 構造を DIF プログラム化するには，当然，制御フラグのスタックが必要となる．PL/DIF コンパイラは，たとえば(b)の構造を，次のようなスタック操作を含む命令系列に変換する．

```
R1←true(cond)      条件判断結果をレジスタ1へ
R2←R1バー          その1の補数をレジスタ2へ
R1←R1・TOP  }
R2←R2・TOP  }      スタックトップとの論理積を求める
PUSH R2     }
PUSH R1     }      else 部，then 部用の制御フラグをスタックへプッシュ
stmt1
POP
stmt2
POP
```

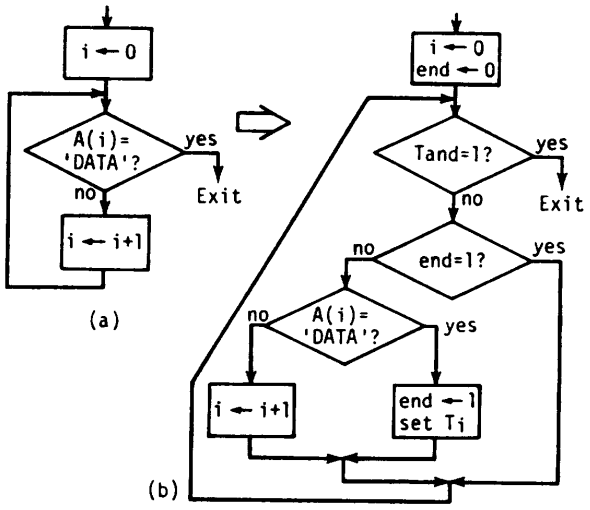


図6 loop 構造の DIF プログラムへの変換
Fig. 6 Conversion of loop-structure into DIF program.

すなわち，if 構造内の各文は，常にスタックトップを制御フラグとして用いるマスク法によって DIF プログラムへ変換される．

4. 3次元画像表示アルゴリズムの並列処理

3次元画像表示に必須な，隠面消去，付影処理には種々の手法があるが，われわれは，隠面消去には古典的な scan-line アルゴリズム[2]，付影処理には shadow polygon の概念[7]を用いるアルゴリズムを採用した．この手法は，凸多角形で構成された立体に対して，隠面消去と付影とが一括して行えるという特徴をもつ．

G-PSYCO は，ホストプロセッサで生成された立体モデル構造を受け取り，これに，**図7**に示す画像表

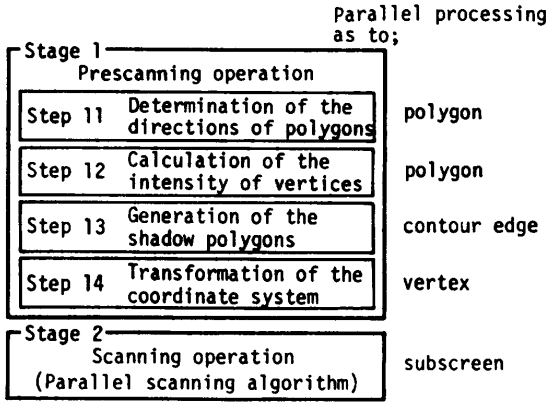


図7 3次元画像表示アルゴリズム
Fig. 7 Algorithm for generating three-dimensional images.

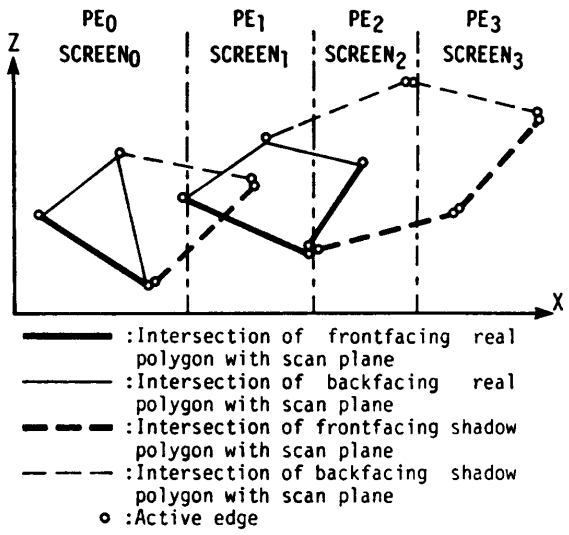


図 8 並列スキャニングアルゴリズムにおける active edge の分割
Fig. 8 Division of active edges for parallel scanning algorithm.

示アルゴリズムを施す．このアルゴリズム自体の詳細はすでに発表済みであるので[9),10)]，本稿では PSYCO による並列処理手法を中心に簡単な説明にとどめる．

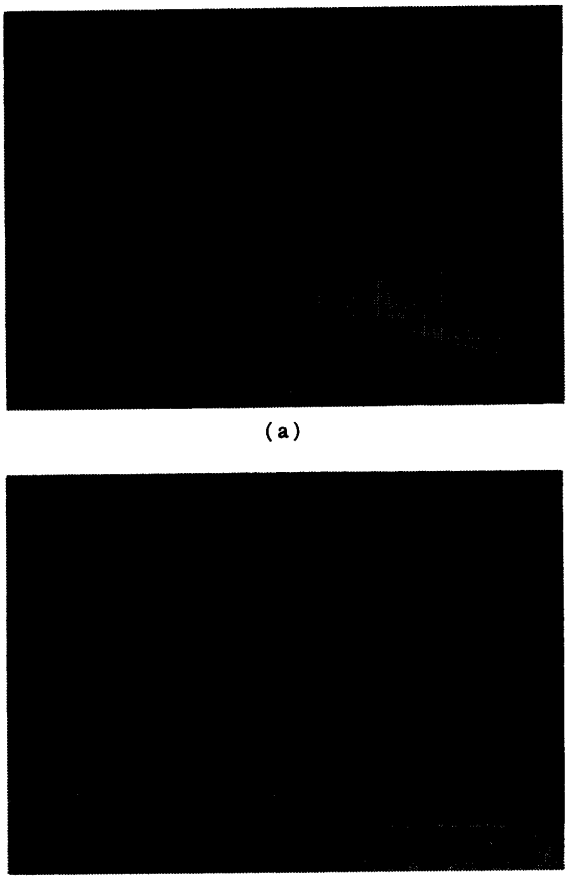
ステージ1は，図7に示すようにスキャニングの前処理を行う四つのステップから成り，これらは，互いに独立な SIMD 型処理に分解できる．

ステップ11では，立体を構成する各平面が視点/光源に表向きか裏向きかを，平面の法線ベクトルと視点/光源へのベクトルとの内積の正負で判定する．平面に関する処理は互いに独立であるため，PSYCO の各 PE に平面を一つずつ与えて並列処理させる．

ステップ12では，簡単な式により平面頂点の輝度を算出する．平面内の他の点の輝度はステージ2で内挿により求める．このステップでも，各 PE に平面，光源のデータを与え，面単位の並列処理を行わせる．

ステップ13では，付影に必要な shadow polygon を生成する．shadow polygon とは，立体の輪郭線を光源に関して投射することにより得られる仮想の平面である．これに対し，立体を構成する実際の平面を real polygon と呼んでいる．1本の輪郭線から一つの shadow polygon が生成されるので，各 PE に輪郭線を与え，線単位の並列処理を行わせる．

ステップ14は，ここまでの処理が施された原座標系から，透視図を得るためにスクリーン座標系へ変換する過程であり，立体各頂点座標に同一の変換マトリクスを乗ずるという，点単位の並列処理を行う．



(a)

(b)

図 9 3次元画像表示例
Fig. 9 An example of three-dimensional color image.

ステージ2は，スクリーン座標系において，スキャン平面と呼ぶ水平面と，real polygon や shadow polygon を構成する稜との交点を求め，その深さ（Z の値）を比較することによって隠面消去と付影を行い，画面全体をこのスキャン平面で走査して最終的に色彩画像を得る，最も重要な段階である．

このステージを並列処理する手法として，われわれは図8に示す並列スキャニングアルゴリズムを開発した．すなわち，あるスキャン平面と交点をもつ稜 (active edge) の個数を PE 台数で除して画面を分割し，小画面を各 PE に割り当て，それぞれにスキャニング処理を行わせる．この処理の結果，各 PE は小画面内のスキャンライン上の輝度を算出し LS に置く．コントロールプロセッサはそれらを読み出し，画像メモリの適切な位置に書き込む．PSYCO が一つのスキャン平面を処理している間に，コントロールプロセッサは次のスキャン平面に関する準備を行うとい

う，2段階のパイプラインが実現されている．

図9に画像表示例を示す．これらの例では，PSYCOのPEの台数を8に設定したG-PSYCOが，ホストプロセッサから立体モデルを受了後画像表示を終了するまでに約1分を要した．

5. G-PSYCOの性能評価

5.1 PSYCOの並列処理性能

SIMDマシンの処理性能は，PEの台数Nに比例せず$\log_2 N$に比例することがMinskyの経験則として知られ，Flynnはその原因として次の4点

（1） PE間通信に伴うオーバヘッド

（2） ベクトルの個数とPE台数との不整合

（3） PEアレイへのデータ転送，並列計算の準備等の並列化不可能な処理

（4） if構造による性能劣化

を挙げ，(4)を主要因としてMinskyの経験則を説明している[4]．しかし，この解析は図3に示したSIMDマシンの一般型に基づいており，PSYCOには適用できない．以下に，DIFプログラムの構造に基づき，PSYCOの並列処理性能について考察する．

（a） if構造

PEの状態によりif構造内のパスの実行を動的に選択する従来の方式と異なり，PSYCOでは全PEがif構造内の全パスを実行する．いくつかの基本算術演算ルーチンについて，if構造のみ用いたDIFプログラムと通常プログラムとの実行時間長比較を図10に示す．ただし，通常プログラムにおいては各分岐確率が等しいと仮定して実行時間を見積もっている．この図で示されるように，DIFプログラムの効率は決してよいとはいえないが，その実行時間長は全PE間で同一である．したがって，コントロールプロセッサによるデータの分配，結果の収集に伴うオーバヘッドを無視すれば，if構造の並列処理性能$P_{if} \propto N$が成立する．

	type of computation	DIF program	ordinary program
F type	addition	40	6
	subtraction	40	6
	multiplication	15	9
	division	21	18
	square root	142	130
I type	addition	0.3	0.3
	subtraction	0.6	0.6
	multiplication	19	11
	division	31	28
	square root	300	284

(X1000 machine states)

F type:Floating point number

I type:Fixed point number

図10 DIFプログラムと通常のプログラムとの実行時間比較

Fig. 10 Comparisons of execution time between ordinary program and DIF program.

（b） loop構造

前述したようにPSYCOにおいては，あるloop構造は，最大loop回数を必要とするデータを担当したPEの処理終了を他のPEが待つ形態で並列処理される．全データに対し必要なloop構造の個数をMとし，処理時間長がxであるloopの個数$n(x)$が正規分布をなすと仮定し，また，loop構造の処理負荷がPE間に最適に分散されたと仮定すれば，PSYCOによるloop構造の処理は，図11に示す状況であると考えることができる．すなわち，各loop構造をその処理時間に比例する長さの実線で表し，これらをその長さでソートすれば，実線端点を結ぶ曲線は，正規分布のc.d.f.（累積分布関数）に一致する．図11で，幅Nの1区画が1回の並列処理に相当し，破線は，その際に最長のloop構造処理終了を待つための無効処理を表す．したがって，曲線左側の実線部分の面積S_1が必要な総処理時間を，破線部分の総面積S_2が総無効処理時間を，それぞれ表すことになる．1台のPEによる性能$1/S_1$に対するN台のPEによる並列処理性能比P_{loop}は

$$P_{loop}=N\cdot S_1/(S_1+S_2)$$

で表され，正規分布のc.d.f.曲線の点対称性から

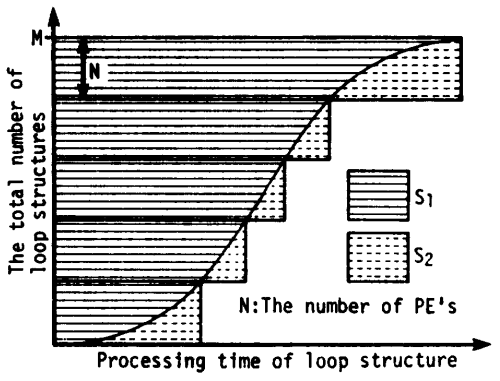


図11 PSYCOによるloop構造の理想的な並列処理

Fig. 11 Parallel processing scheme of PSYCO for loop structures.

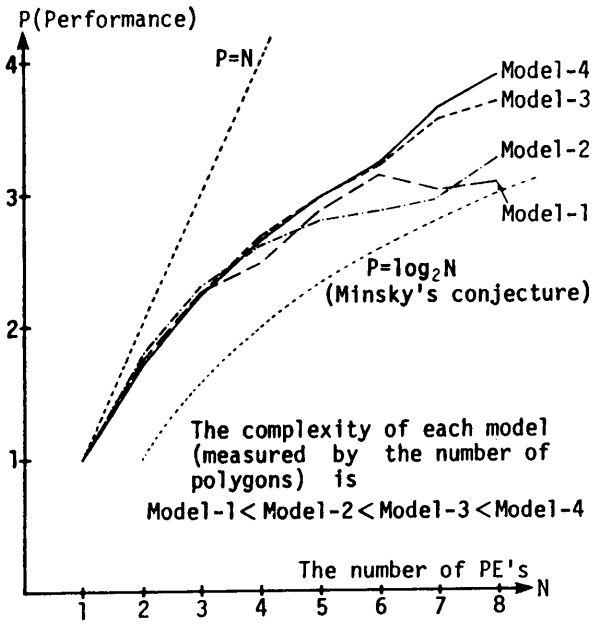


図 12 G-PSYCO の並列処理性能
Fig. 12 Parallel processing performance of G-PSYCO.

$S_1/(S_1+S_2)=M/(M+N)$

が得られるので，次式が成立する．

$$P_{loop}=N/(N/M+1) \qquad (1)$$

5.2 3次元画像表示処理における並列処理効果

ホストプロセッサから立体モデルを受了後画像表示を終了するまでに要する G-PSYCO の処理時間の実測値をいくつかのモデルについて求め，これから，各モデルの PE 1台による処理速度が1となるように正規化して得られた並列処理性能を，**図12**に示す．

全処理時間の約 4/5 は，ネストした loop 構造を多数含むスキャニング処理が占めている．G-PSYCO においては，Flynn の挙げた4項目のうち，(1)を無視でき，また，並列スキャニング処理におけるパイプライン処理により(3)の要因もほとんど無視できると考えれば，おもに loop 構造が並列処理性能に関与していると見なすことができる．

並列スキャニング処理において，コントロールプロセッサは，loop 処理の負荷予測に基づく負荷分散を行うのではなく，active edge の個数に基づいて画面を単純に分割し，小画面の処理全部を PE に分担させて，並列処理のスケジューリングを簡単化している．このため，loop 処理の負荷が PE 間に均等分散されるという仮定に基づく式(1)をそのまま適用できない．とくに，Model-1 の例は，N の増加による画面分割の変化に伴い，負荷が特定 PE に集中し，性能の低下を招いたと思われる．しかし，Minsky の経験則より良好な性能を示し，また，立体モデル構造の複雑化につれてMが大となり性能が改善されるという図12の傾向は，式(1)から説明されると考えることができる．

また，PSYCO では，LS 内にプログラムを置くことも可能である．LS 内に置かれた loop 処理をプロセスとし，プロセス間同期制御プログラムを CS 内に置いて，プロセス終了ごとに同期させる方式を採れば，無効処理時間の全体に占める比率が減少し，式(1)により近い性能を示すことが期待できる．

6. むすび

3次元カラーグラフィックスシステムにおける画像生成処理は，G-PSYCO のような高機能端末によって行うのが，親計算機の負担を軽減する点で望ましいシステム形態であると考えられる．本稿では，3次元画像生成処理に含まれる SIMD 型処理に対しては，DIF プログラム概念の採用によって非常に簡単なハードウェア構成で SIMD マシンを実現できることを示した．この方式は，市販のマイクロコンピュータボードに些細な改善を施すだけで容易に SIMD マシンを開発でき，その保守性，拡張性も優れているので，単純な SIMD 型処理を含む他の分野にも十分適用できると考える．また，特殊なハードウェアによる低レベル並列マシンではなく，マイクロプロセッサレベルの並列マシンであるため，前述したプロセス同期型制御も可能であり，並列処理のレベルに柔軟性があるといえる．

われわれの用いた並列スキャニングアルゴリズムでは，画面を分割して PE に分担させるという単純な並列処理方式をとり，負荷の均等分散を十分考慮していないため，顕著な並列処理効果が得られたとはいいがたい．今後，PSYCO に適したより効率的な隠面消去手法について検討するとともに，プロセス同期型制御も取り入れて，並列処理性能の改善を図る予定である．

参考文献

1) Flynn, M. J. : Very High-Speed Computing Systems, *Proc. IEEE*, Vol. 54, No. 12, pp. 1901-1909 (1966).
2) Bouknight, W. J. : A Procedure for Generation of Three-Dimensional Half-Toned Compu-

ter Graphics Presentations, *CACM*, Vol. 13, No. 9, pp. 527-536 (1970).
3) Bouknight, W. J. : The ILLIAC Ⅳ System, *Proc. IEEE*, Vol. 60, No. 4, pp. 369-388 (1972).
4) Flynn, M. J. : Some Computer Organizations and Their Effectiveness, *IEEE Trans. Comput.*, Vol. C-21, No. 9, pp. 948-960 (1972).
5) 岡田, 田島, 森：新しい可変構造プロセッサアレイ—ETL ANMA, 電子通信学会技術研究報告, EC76-81 (1977).
6) Flanders, P. M. et al. : Efficient High Speed Computing with the Distributed Array Processor, *High Speed Computer and Algorithm Organization*, pp. 113-128, Academic Press, New York (1977).
7) Crow, F. C. : Shadow Algorithm for Computer Graphics, *SIGGRAPH-ACM*, Vol. 11, No. 2, pp. 242-248 (1977).
8) Batcher, K. E. : Design of a Massively Parallel Processor, *IEEE Trans. Comput.*, Vol. C-29, No. 9, pp. 836-840 (1980).
9) Kubo, M. et al. : Multi-Microprocessor System for Three-Dimensional Color Graphics, Proc. IFIP 80, pp. 145-150 (1980).
10) Kubo, M. et al. : A Parallel Processor System Dedicated to SIMD and Its Application to Three-Dimensional Color Graphics, Proc. 3rd Int. Conf. on Distributed Computing Systems, pp. 870-875 (1982).

(昭和58年3月7日受付)
(昭和58年6月20日採録)