

低レベル並列処理計算機による3次元色彩図形表示処理†

湯 浅 真 治** 中 田 登 志 之*** 新 實 治 男****
富 田 眞 治**** 萩 原 宏****

我々はコンピュータの基本構成方式として低レベル並列処理方式を提案し、大規模試作システム QA-2 を開発してきた。低レベル並列処理方式とは、比較的長い命令の異なるフィールドで多数の演算器 (ALU)、メモリやバスなどを同時に制御する方式である。VLSI 時代を迎えて、個々の応用分野に対して、各種の並列処理方式の有効性と限界を明確にすることが重要な課題となっている。本論文は、3次元色彩図形表示処理に対する低レベル並列処理方式の有効性と限界を実験機 QA-2 を用いて示したものである。QA-2 の実装 ALU 台数は4台が妥当であり、これ以上 ALU を増設しても有意な性能向上が望めないことを示した。また、ALU 連鎖演算方式などの逐次処理支援機構、高度な順序制御方式などのシステム性能に及ぼす効果について言及し、単純なステップ数で比較した場合、IBM 370 の1/5程度に動的ステップ数が減少することなどを示した。

1. はじめに

コンピュータ・グラフィックス (CG) は、人間と計算機とのマン・マシン・インタフェースとして極めて重要な技術であり、その一層の普及のためには、コスト/パフォーマンスのよいシステムの開発が必要不可欠である。実際、近年の VLSI 技術の発展を背景にして、3次元図形の高速度表示を目的としたパイプライン方式やマルチプロセッサ方式などによる専用システムの開発が各所で進められている¹⁾。しかし、3次元図形の表示のためには、座標変換のような大量均質データの処理と、隠れ面消去等の複雑な処理という、性質の異なる処理過程を双方ともに効率よく実行する必要がある。従来の専用システムではこれら二つの処理過程を分離し、その双方あるいは一方のみを適当な専用化手法により高速化するものがほとんどであった。

これに対して、我々は計算機の基本構成方式として低レベル並列処理方式を提案し、同方式による大規模試作システム QA-2²⁾ を用いて、3次元色彩図形表示処理に対する応用実験を行った。この方式は、図1に示すように比較的長い命令の異なるフィールドで多数の演算器、メモリやバスなどを同時に制御する方式であり、処理アルゴリズムに内在する並列処理可能性にそれほど依存することなくハードウェアの持つ並列処

理性能を良く引き出すことができる。したがって、大量均質データの処理にも複雑な制御を要する処理にも同程度の並列処理効果を発揮でき、単一のハードウェアによる効率のよいシステムの構築が期待できる。

本論文は、3次元色彩図形表示処理に対する低レベル並列処理方式の有効性と限界を実験機 QA-2 を用いて示したものであり、演算器 (ALU) の実装個数、逐次処理支援機構、順序制御方式などがシステム性能に及ぼす効果について考察した。

2. 低レベル並列処理方式と試作機 QA-2 の構成

2.1 低レベル並列処理方式

低レベル並列処理方式⁴⁾ は、コントロール駆動方式に属する「スカラ並列演算方式」³⁾ の一種であり、図1に示したように、比較的長い命令を多数のフィールドに分割し、各々のフィールドで ALU などの機能装置を独立に制御する方式である。この方式を採用したマシンとして、QA-2 のほかに CDC 社の Cyber-plus⁵⁾、Yale 大学の ELI⁶⁾ などがある。低レベル並列処理方式は、

- ① 並列演算の可能性の検出はコンパイル時など、実行前に行うので、制御装置の構成をデータ駆動方式などと比べて単純化できる。
- ② ALU などの機能装置間の通信は命令内のフィールドで直接制御され、通信オーバーヘッドは少ない。

などの特長を有している。ただし、ハードウェア構成上、次のような点の考慮が必要である。

- ① ALU の実装個数と使用率：この方式では、ALU 個数に近い並列度が得られる時に高速処

† 3-D Color Image Generation by a Computer with Low-Level Parallelism by SHINJI YUASA (Toyota Motors Co. Ltd.), TOSHIYUKI NAKATA (NEC Corporation), HARUO NIIMI, SHINJI TOMITA and HIROSHI HAGIWARA (Department of Information Science, Faculty of Engineering, Kyoto University).

** トヨタ自動車(株)

*** 日本電気(株)

**** 京都大学工学部情報工学科

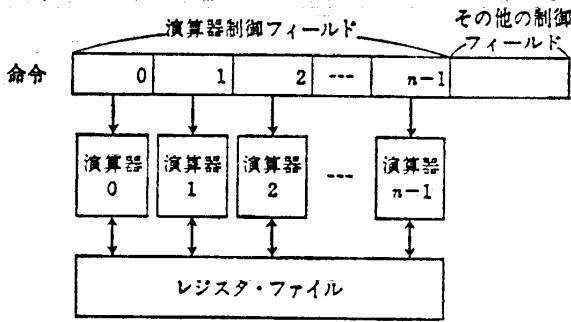


図1 低レベル並列処理方式
Fig. 1 Low-level parallel processing scheme.

理を達成できる。しかし、並列度が低い時には ALU 制御フィールドに遊びができ、命令のビット使用率が低下する。

- ② ALU 連鎖演算方式：各 ALU の入出力ポート間を直接結合する「ALU 連鎖演算方式」を採用することにより、データ依存性によって逐次的な処理が必要な場合でも、ALU の使用率を低下させることが少なくなる。
- ③ 柔軟な順序制御部の構成方式：複数 ALU で多数のステータスが生成されるので、多様なステータスの組み合わせによる条件分岐機構など、

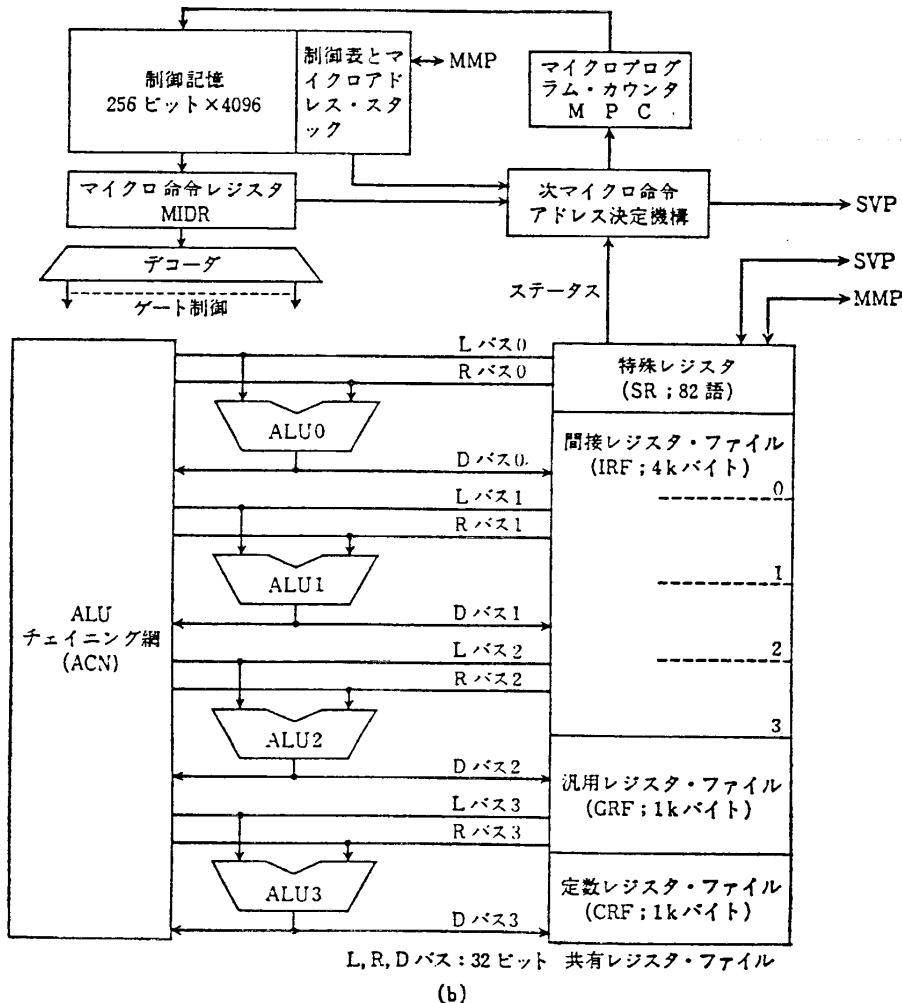
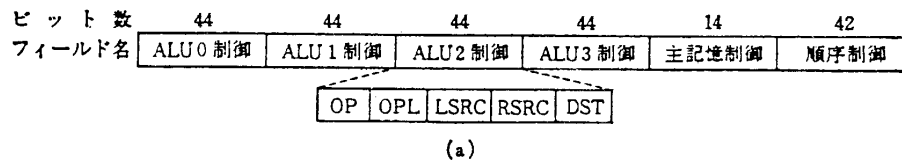


図2 QA-2 のハードウェア構成
Fig. 2 Hardware organization of the QA-2.

順序制御部を柔軟な構成にしておく必要がある。

- ④ 簡潔な並列構造：処理速度の向上はコンパイラの能力に大きく依存するので、コード生成のしやすい簡潔なハードウェア構造が望まれる。

2.2 QA-2 のハードウェア構成²⁾

低レベル並列処理計算機QA-2の特長は以下のよう
にまとめられる。

(i) 同一で高機能な4個のALUによる低レベル並列処理方式：QA-2では、図2に示すように、4個の可変長(1, 2, 3, 4バイト)ALUが1ワード(256ビット)の水平型マイクロ命令の相異なるフィールドで独立に制御され、それらが均一で大容量のレジスタ・ファイル(6kバイト)を共有して動作する。

4個のALUは互いに独立な四つのオペランド群に対して並列演算を実行できるだけでなく、1個の演算結果を他のALU演算の入力オペランドとして使用するALU連鎖演算も可能である。したがって、例えば $F=(A+B)*C+D*E$ は1マイクロ命令で記述できる。また、 $A\sim F$ のすべての変数は共有レジスタ上に何ら制限なく割り付けることができる。

(ii) 高機能順序制御方式：QA-2には、高度な順序制御機能が用意されている。条件分岐の方法に次の二つの形式がある。

```
IF F(A1, A2, ..., Ak) THEN (分岐操作-1)
    ELSE (分岐操作-2)

CASE (A1, A2, ..., Ak) OF /0/ (分岐操作-1)
    :
    /2k-1/ (分岐操作-2k)
```

IF分岐の論理関数 F は、最大8変数 ($1 \leq k \leq 8$) までの論理変数を指定し、ユーザが任意に定義できる(最大256種の論理関数のうちの1種)。またCASE分岐の場合も、8変数までユーザが自由に組合せを定義できる。各分岐操作には、“GOTO”文、“CALL”文、“RETURN”文が、何の制約もなく指定できる。このようにして、多数のステータスに基づいた順序制御の高速化を図っている。

(iii) 高速主記憶アクセス機構：主記憶は16バンク構成になっており、4組の主記憶アクセス・レジスタ群(MDR, MARなど)を用いて、1マイクロ命令で相異なる4個のアドレスにある可変長(1~4バイト)データにアクセスできる(並列アクセス機能)。また複数個のMDRをリンクすることにより、一つのMARで4バイト以上(最大16バイト)のデータへのアクセスも可能である(多倍長アクセス機能)。

データ・チャンネルの速度はバースト・モード時で40MB/sである。図形処理では、主記憶内に設けた画像記憶(512×512 pixels×16 bits)に1画面分のデータを揃えた後、このチャンネルを通して外部のモニタのFB(フレーム・バッファ)に高速転送する。

3. 3次元図形表示システムの概要

3.1 3次元図形表示アルゴリズム⁷⁾

本システムでは、多角形で構成された多面体を処理対象としている(多面体モデル)。曲面は多角形で近似することにより、取り扱っている。多面体モデルは、いろいろな形状の図形を柔軟に扱えるとともに、処理が簡単になるという利点を持つ。

図形表示は、座標変換、クリッピング、隠面消去、シェイディングの過程を経てなされる。3次元図形表示処理のうち、隠面消去処理が全処理量の大部分(我がシステムでは90%以上)を占め、図形表示処理の核となるので、種々のアルゴリズムが提案されている。本システムでは、スキャンライン・アルゴリズム(以下、SLAと略す)とZバッファ・アルゴリズム(以下、ZBAと略す)について検討した。

また、図形表示においては、より現実感を持った画像を生成する必要がある。そこで、本システムでは、現実感表出処理として、付影処理(Shadowing)、スムーズ・シェイディング(Smooth-Shading)、境界線平滑化処理(Anti-Aliasing)の組み込みを図っている。なお、座標変換、クリッピング、隠面消去および上記の現実感表出処理は、すべてQA-2のアセンブリ言語で記述されている。

3.2 SLAによる隠面消去

SLAは、ラスタ・スキャン型ディスプレイにおける水平方向の画素集合である「スキャンライン(Scan-line)」ごとに次のように処理を進める。

図3(a)の X_s - Y_s 平面は表示スクリーンであり、 Z_s 軸は視線の方向を示している。この座標系は「スクリーン座標系」と呼ばれるもので、図3の X_s - Z_s 平面に平行な平面のことを、「スキャンライン平面」と呼び、スキャンライン平面と各多角形との交線(図中のA, Bなど)を「セグメント(Segment)」と呼ぶ。

図3(b)に $Y_s = Y$ のスキャンライン平面で切った場合の X_s - Z_s 平面上のセグメントを示す。隠面消去は、セグメントに重なり(例えばBとD)がある場合、より手前のセグメント(この場合B)を表示するよう処理がなされる。このため、スキャンライン平面

上のセグメントを、その左端点の出現順にソートして結び付けた「アクティブ・エッジ・リスト (Active Edge List, 以下 AEL と記す)」と呼ぶデータ構造が必要である。図3の例では、AELには $A \rightarrow D \rightarrow B \rightarrow E \rightarrow C$ の順にセグメントのデータが結び付けられる。隠面消去は AEL をたどりながら行う。スキャンラインを一つ次に進める場合 ($Y_s = Y_s + \Delta Y_s$) には、AEL の更新が必要であるが、スキャンライン間の類似性と多面体モデルの特徴を利用して比較的簡単に行うことができる。

本アルゴリズムにおける代表的な処理ルーチンとして以下のものがある。

(i) SCAN ルーチン: AEL の更新と可視セグメント決定のための主制御ルーチンであり、QA-2 の主記憶並列アクセス機能、および強力な順序制御機能を有効に利用し、AEL の逐次探索を効率的に行っている (4.3 節(ii)参照)。

(ii) WCLOR ルーチン: 複数画素に同一色を書き込むルーチンであり、QA-2 の主記憶多倍長 (最大 16 バイト) アクセス機能を用いて高速化を図っている (4.3 節(ii)参照)。

(iii) DIVIDE ルーチン: dx_s/dy_s などの計算のための除算ルーチンである。引き放し法における複数の場合分けされた計算過程を、複数の ALU で並行して同時に演算することで、高速化を達成している (4.3 節(iii)参照)。

(iv) SORT ルーチン: AEL などのソーティングを行う。ソートすべきデータを QA-2 の高速大容量のレジスタ・ファイル (IRF) 上にロードし、その上でソーティングを行うことにより、主記憶アクセスを大幅に減少させ、高速化を達成した (4.3 節(i)参照)。

(v) その他: 座標変換、クリッピング処理など。

また、現実感表示処理については、付影処理は仮想的な影多角柱を生成する方法、スムーズ・シェイディングは Gouraud の方法、境界線平滑化処理は稜線の傾きから色の混合比を求める方法により、それぞれ、SLA のわずかな変更で実現した。

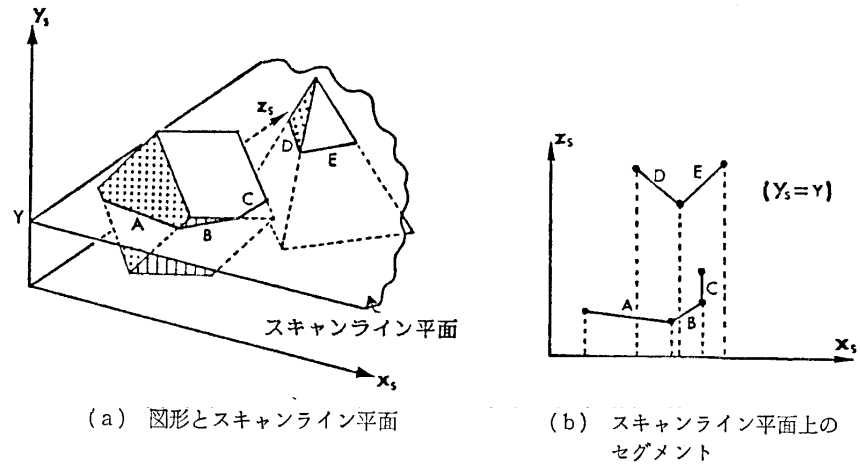


図3 スキャンライン・アルゴリズム (SLA)
Fig. 3 Scan-line algorithm (SLA).

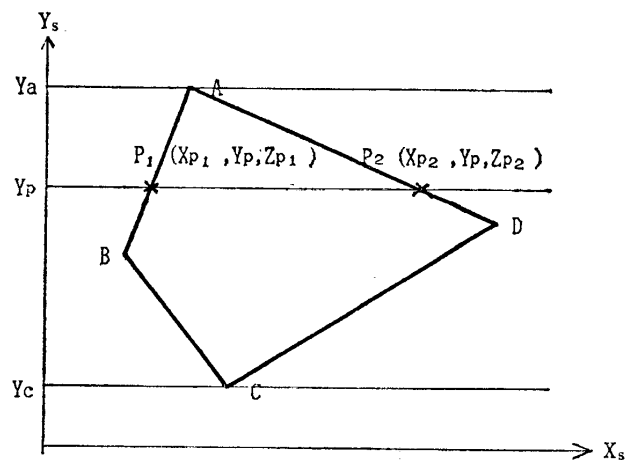


図4 Zバッファ・アルゴリズム (ZBA)
Fig. 4 Z-buffer algorithm.

3.3 ZBA による隠面消去

ZBA は、画像記憶とは別に、画素ごとの Z 座標値を保持する Z バッファを用いて隠面消去を行う方法である。ZBA は、アルゴリズム自体が単純であり、高速な図形表示を行える可能性がある。

図4に示すように、多角形の頂点 (A, B, C, D) を抽出した後、Y_s 座標値でソートし、Y_s 方向に順に多角形とスキャンラインとの交線線分を求める。そして各線分 (例えば図4の P₁P₂) 内の画素ごとに Z 座標値を求め、各画素の Z バッファ値と比較する。Z バッファ値の方が大きいときには Z バッファ値を新しい Z 座標値で置き換え、対応する画像記憶の画素を書き換える。この操作をすべての多角形について繰り返す。本アルゴリズムにおける代表的な処理ルーチンに以下の

ものがある。

(i) ZMAIN ルーチン: 画像記憶, Zバッファの初期化と, 多角形の頂点データの読出しなどの制御を行う。

(ii) LINE ルーチン: 稜線の両端点から輪郭点(例えば図4の P_1, P_2)を抽出する。

(iii) ZWRITE ルーチン: LINE ルーチンで求めたデータを用い, Z座標値を求め, 画像記憶とZバッファを更新する。この比較・更新の処理を4画素同時に並行して行っており, QA-2の多方向分岐機能と主記憶並列アクセス機能が効率良く働いている。このルーチンでの処理時間は全体の60%以上を占めており, QA-2の低レベル並列処理方式が特に寄与している部分である(4.3節(i)参照)。

(iv) DELTA ルーチン: dz_x/dx , などの増分値を計算する。

(v) DIVIDE ルーチン: SLAの場合と同様, 引き放し法による除算を行う。

(vi) その他: 座標変換, クリッピング処理など。

現実感表出処理については, スムーズ・シェイディングはSLAの場合と同様のGouraudの方法により実現している。しかし, SLAの場合と同様の境界線

平滑化処理は, ZBAの性質から組み込みは困難である。また付影処理は, Zバッファと同容量の影バッファを用意することで実現できるが, 本システムでは, QA-2の主記憶容量の制限のため, 行っていない。

4. システムの性能評価

4.1 スキャンライン・アルゴリズムの場合

(i) 図形データ量と処理量の関係

ランダムに発生させた立方体を図形データとして, 図形データ量(立方体数 n)と処理ステップ数の関係を求めた。結果を図5に示す。棒グラフは現実感表出処理を行わない場合である。QA-2の処理能力は, 100万ステップ/秒である。図5より, 立方体約30個を1秒で表示できることがわかる。また, 処理内容別でみると, 例えば立方体数30の場合, 隠面除去の主制御(SCAN, 31.8%), AELなどのソート(SORT, 40.0%)の割合が高くなっている。立方体数が増加するにつれてSORTの割合が大きくなっている。総実行ステップ数は近似的に

$$265n^2 + 13300n + 205000$$

と表すことができる。 n^2 の項は, AELのソーティング, およびAELにおける同一多角形に属するエッジ

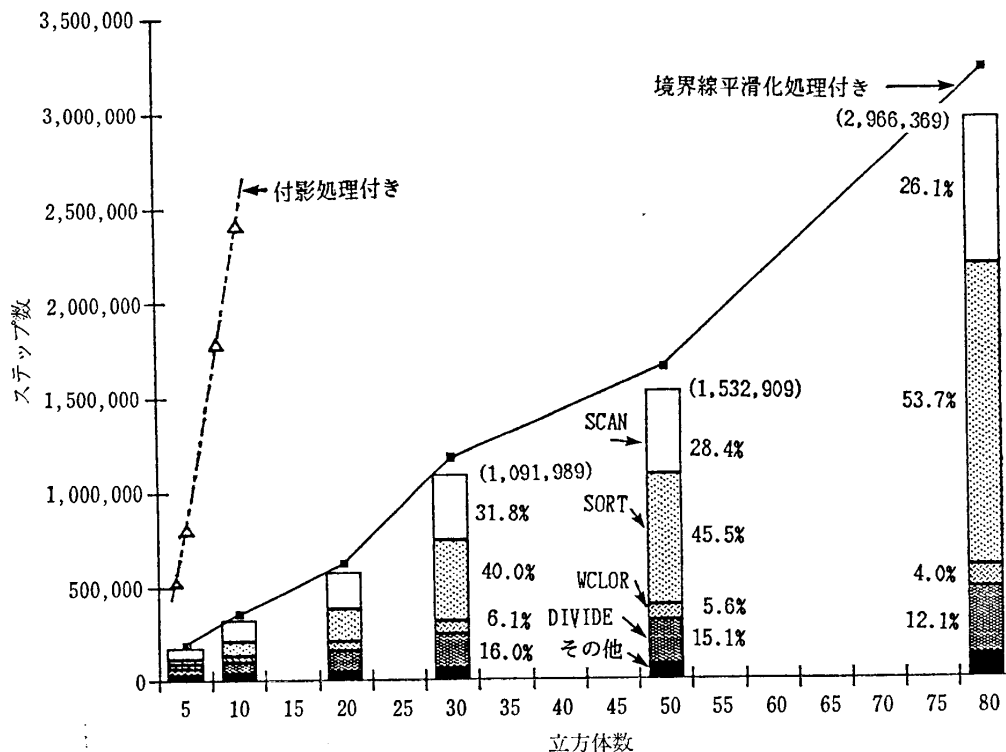


図5 SLAによる処理ステップ数
Fig. 5 Processing steps by the SLA.

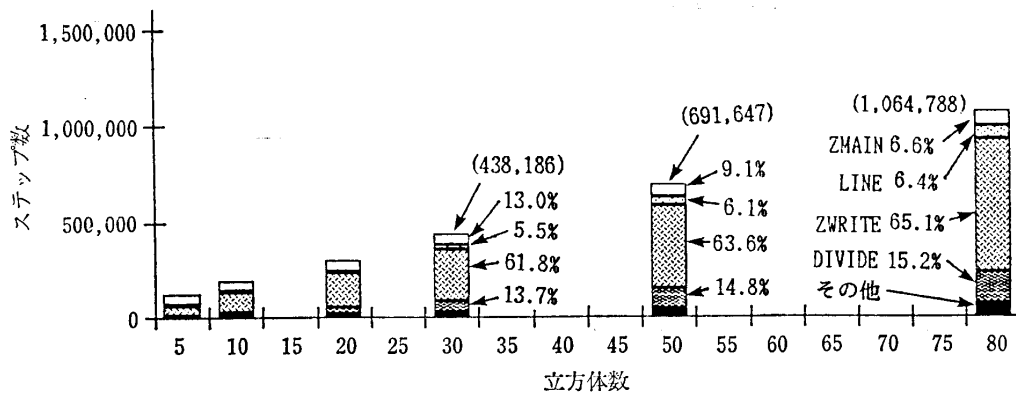


図6 Zバッファ・アルゴリズムによる処理ステップ数
Fig. 6 Processing steps by the Z-buffer algorithm.

の探索によるものである。AEL 中では並べ替えが少ないという性質があるため、ソーティング・アルゴリズムとして、バブル・ソートを採用した。クイック・ソートを用いると、ステップ数は逆に、約12%の増加となる。

(ii) 現実感表出処理の負荷

① 付影処理

その内部にある物体に影を落とす働きを有する影多角柱を求める。影多角柱を構成している影多角形を実面の多角形と同様に扱って隠面消去を行い、影をつけるかどうかを判定する。処理すべき多角形に影多角形が追加されるとともに、相貫するセグメントも増加するので、全体のステップ数は急増する。図5の立方体の場合には、約6倍のステップ数に増加している。

② スムーズ・シェイディング

頂点ごとに求めた輝度から、多角形内の各画素の輝度を線形補間する方法(Gouraudの方法)で実現した。多面体の場合は、この処理は必要ないので WCLOR ルーチンで8画素分を1ステップで書き込める。しかし、曲面の多角形近似では、この処理のために、1ステップで1画素の書き込みとなる。円柱を12角柱で近似した図形でテストした結果、スムーズ・シェイディングを行った場合、処理量は約30%増加した。

③ 境界線平滑化処理

境界線の傾きから色の混合比を求める方法で実現した。スキャンラインと境界線の交点1個につき、約20ステップを要し、図5の立方体の場合には、10%程度のステップ数の増加で実現できた。

4.2 Zバッファ・アルゴリズムの場合

(i) 図形データ量と処理量の関係

SLA の場合と同様に、ランダムに発生させた立方

体を図形データとして、図形データ量と処理ステップ数との関係を求めた(図6参照)。この場合では、1秒間に約80個の立方体を表示することができる。処理内容でみると、Zバッファの比較更新を行う ZWRITE ルーチンが60%以上を占めることがわかる。また、マイクロプログラムのフロー解析を行ったところ、処理ステップ数は、立方体数に線形に比例して増加することが確かめられた。実行ステップ数は近似的に

$$12500n + 64300$$

と表すことができる。

(ii) 現実感表出処理の負荷

スムーズ・シェイディングは、SLA の場合と同様、Gouraudの方法で行った。すなわち、頂点ごとに求めた輝度から、Z座標値と同様、線形補間して各画素の輝度を求めている。SLAと同様に、12角柱で近似した円柱でテストした結果、約16%の処理ステップ数の増加で実現できることが確かめられた。しかし、ZBAでは図形相互の位置関係を全く意識していないため、Z値の比較で可視と判定される度にスムーズ・シェイディングを行う必要がある。したがって、図形のZ方向の重なりが増加すると、最終的に表示されない面に対してシェイディング計算を行う可能性が高まり、SLAの場合とは異なり、無駄な処理が多くなる。この無駄を回避し、最後に一度だけ可視多角形に対してだけシェイディング計算を行うためには、Zバッファとは別に、各画素ごとに可視面を記憶するバッファが必要となるばかりでなく、すべての多角形データを処理の最後まで保存しておく必要がある。このため、Zバッファ法の簡潔さが失われる。

表 1 主なルーチンの評価データ (立方体 30 個の場合)
Table 1 Evaluation data of major routines (in case of 30 cubes).

ルーチン名		スキャンライン・アルゴリズム				Zバッファ・アルゴリズム	
		SCAN	WCLOR	SORT	DIVIDE	ZWRITE	
動的ステップ数		346,775	66,199	437,245	138,174	270,836	
動的ステップ数の割合		31.8%	6.1%	40.0%	16.0%	61.8%	
RALU	平均 ALU 使用個数	3.04	2.56	3.07	3.16	3.82	
	ALU 連鎖演算の割合	7.4%	39.0%	21.2%	36.5%	19.2%	
MMP	平均主記憶参照回数	1.17	0.34	0.14	0.00	0.59	
	制御フィールドの個数	1	34.2%	100.0%	28.5%	0.0%	3.1%
		2	31.8%	0.0%	71.2%	0.0%	71.7%
		3	12.4%	0.0%	0.0%	0.0%	0.0%
		4	21.6%	0.0%	0.0%	0.0%	25.0%
SCU	無条件分岐の割合	66.1%	57.1%	43.8%	3.9%	66.3%	
	2方向分岐の割合	24.5%	42.9%	52.5%	3.9%	20.4%	
	多方向分岐の割合	9.4%	0.0%	3.7%	92.2%	13.3%	
	条件分岐のうちカレント・モードを用いる割合	68.0%	1.9%	41.4%	100.0%	30.8%	

4.3 QA-2 の評価

(i) ALU 使用率と ALU 実装個数の妥当性

主なルーチンにおける評価データを表 1 に示す。平均 ALU 使用個数とは 1 マイクロ命令で使用される ALU 演算個数の平均値である。

SLA の場合、平均 ALU 使用個数は画像記憶に色を書き込む WCLOR ルーチンを除いて、3 以上となっている。特に、動的ステップ数の多い SORT ルーチンで 3.07 と高い値となっている。先に述べたように、隠面消去ではアクティブ・エッジ・リスト(AEL)のソートが必要である。AEL は主記憶上にリストの形式で格納されている。リストを逐次的にたどりながらソートするのはオーバーヘッドが大きいので、本システムではいったんリスト要素の主記憶内アドレスとソートすべき値をペアにして大容量レジスタ・ファイル(IRF)に格納し、IRF 上で四つの ALU を利用して、ソーティングを高速に行っている。

ZBA の ZWRITE ルーチンでは、平均 ALU 使用個数が 3.82 と、4 に近い値を示している。これは、アルゴリズム自体の持つ並列性に加えて、QA-2 の並列性を高める機能が有効に働いた結果であると考えられる。実際の処理では、ALU 連鎖演算機能を用いた

表 2 修正アーキテクチャとの性能比較 (立方体 30 個の場合)
Table 2 Performance comparison with modified architectures (in case of 30 cubes).

	ルーチン名	動的ステップ数	ALU 連鎖演算機能がない場合	順序制御が単純な場合
SLA	SCAN	346,775	416,237(1.20倍)	495,642(1.43倍)
	WCLOR	66,199	82,708(1.25倍)	89,518(1.35倍)
	SORT	437,245	559,278(1.28倍)	797,517(1.82倍)
	DIVIDE	138,174	268,222(1.94倍)	319,240(2.31倍)
ZBA	ZWRITE	270,836	284,612(1.05倍)	364,761(1.35倍)

増分計算により、4 画素分の Z 座標値を求め、Z バッファ内の値と並列比較し、その結果に応じて、多方向分岐を行い、画像記憶と Z バッファの更新を行っている。

ALU 連鎖演算の割合は、全 ALU 演算個数に対する、連鎖演算を行っている演算個数の割合であり、SLA、ZBA とともに平均的にみて 20% 前後の値となっている。ALU 連鎖演算機能を装備していない場合、ステップ数がどの程度増加するかを表 2 に示す。

QA-2 の ALU 実装個数 (=4) が妥当な値であるかどうかを調べるため、ALU の個数と必要ステップ数の関係を求めた。ただし主記憶アクセス・フィールド

表3 ALU 個数とステップ数の関係 (立方体30個の場合)
Table 3 Relationship between the number of ALU's and processing steps (in case of 30 cubes).

ALU 個数		1	2	4	8	16
SLA	SCAN	1,054,245 (3.04)	576,686 (1.66)	346,775 (1.00)	302,291 (0.87)	282,842 (0.82)
	WCLOR	169,361 (2.56)	109,871 (1.66)	66,199 (1.00)	49,690 (0.75)	44,363 (0.67)
	SORT	1,344,423 (3.07)	728,572 (1.67)	437,245 (1.00)	299,401 (0.68)	280,297 (0.64)
	DIVIDE	436,447 (3.16)	268,220 (1.94)	138,174 (1.00)	138,174 (1.00)	138,174 (1.00)
ZBA	ZWRITE	1,033,599 (3.82)	520,659 (1.92)	270,836 (1.00)	198,031 (0.73)	165,060 (0.61)

は ALU 実装個数と同数であるとしている。表3に示すように、ALU 個数を倍の8にしても、必要ステップ数は20%程度しか減少せず、必要なハードウェア量を考慮すると、QA-2のALU 実装個数は妥当な値といえる。

(ii) 高速主記憶アクセス機構

表1の平均主記憶参照回数とは、1マイクロ命令内の四つの主記憶制御フィールドの利用個数の平均を表す。制御フィールドの個数は主記憶アクセスの際の独立した主記憶アクセスの個数を示す。QA-2では大容量レジスタが用意されており、主記憶参照回数は低い値となっている。しかし、SCAN ルーチンでは、主記憶上のリスト構造をたどる処理が多く、1マイクロ命令当たり平均1.17回の主記憶アクセスを行っている。四つの独立したアドレスへのアクセスが21.6%もあり、並列アクセス機能が生かされているといえる。WCLOR ルーチンでは8画素分同一色を書き込む際に、制御フィールド一つを使用して多倍長(16バイト)アクセスを行っている。

(iii) 高機能順序制御方式

表1に示すように、リスト構造を扱う SORT ルーチン、逐次的な処理内容を持つ DIVIDE ルーチンにおいて、条件分岐の割合が極めて高くなっている。DIVIDE ルーチンでは、3変数による多方向分岐機能を利用して、1ステップで1ビットの解を求めている。

QA-2は高度な順序制御機構を有しており、その効果をみるために、

IF A₀ THEN GOTO LABEL (ELSE NEXT) という単純な順序制御だけしか用意されていない場合との比較を行った(表2参照)。その結果、処理ステップ数が1.4~2.3倍になることから、QA-2の有する順序制御機能が処理の並列性を高め、処理ステップ数の減少に非常に貢献していることがわかる。

(iv) IBM 370 との性能比較

表4 QA-2のマイクロ命令のステップ数とIBM 370の機械語のステップ数の比較(立方体30個の場合)

Table 4 Performance comparison with the QA-2 and IBM 370 (in case of 30 cubes).

	ルーチン名	静的ステップ数		動的ステップ数	
		QA-2	IBM 370	QA-2	IBM 370
SLA	SCAN	127	530(4.2倍)	346,775	1,165,918(3.36倍)
	WCLOR	9	46(5.1倍)	66,199	340,593(5.14倍)
	SORT	22	97(4.4倍)	437,245	2,195,933(5.02倍)
ZBA	ZWRITE	35	200(5.7倍)	270,836	1,651,363(6.10倍)

表4に、IBM 370との比較を示す。表4のIBM 370のステップ数を算出する際には、汎用大型コンピュータは命令パイプライン制御方式を採用しており、多くの単純な機械命令は1ステップで実行されるものと仮定している(Load Multiple 命令などの多倍長データ転送命令では8バイトごとに1ステップ必要である)。しかし、命令不在/データ不在によるパイプラインの乱れの影響は、表4の動的ステップ数には含まれていない。

IBM 370のステップ数がQA-2のステップ数と比べて増加する要因として、①IBM 370が1アドレス方式であり、レジスタ数が限られることによりデータ転送関係の命令が増加すること；②分岐に関する独立した命令が必要であること；③カウンタ制御など、ステータス・セットのための命令が必要であること；等が考えられる。QA-2では、①については大容量レジスタで、②、③については水平型マイクロ命令の並列操作で吸収されている。

逆に、IBM 370のステップ数が減少する要因としては、主記憶アクセスにおいてMARのセットが不要であることが考えられる。

比較の結果、主記憶参照率の高いSCANルーチンを除いて、IBM 370のステップ数がQA-2の5倍以上になっている。この数字は、QA-2の高機能性を示す一つの指標になると考えられる。

5. おわりに

本論文では、3次元色彩図形表示処理に対する低レベル並列処理方式の性能評価について述べた。

ALU 実装個数を20~30とするシステムがYale大学のELIプロジェクトで提案されている⁶⁾。分岐命令の分岐方向に偏りがある科学技術計算の分野では、Trace Schedulingと呼ぶ最適化手法が適用でき、20~30台のALUの同時動作が可能であると報告されている。しかし、3次元図形表示処理では、リスト構造を扱う関係から逐次処理部分の占める割合が大きく、このような最適化手法をそのまま適用することはできない。ALU個数は本文で示したように、4程度が適当であると考えられる。3次元図形表示は画面分割などにより、プロセッサ・レベルの並列処理が容易である。VLSI時代を迎えて、並列処理方式の研究は一層重要性を増しており、各種の並列処理方式の有効性と限界を明確にすることが焦眉の課題となっている。今後は、低レベル並列処理方式を採用したプロセッサを多数配置した並列処理方式と、専用プロセッサ、ベクトル・プロセッサ、マルチ・マイクロプロセッサなどの諸方式との性能比較が必要とされよう。

参考文献

- 1) 新實, 富田: グラフィックスとその専用マシン, *bit*, Vol. 17, No. 10, pp. 51-73, 共立出版, 東京 (1985).
- 2) Tomita, S. et al.: A User-Microprogrammable, Local Host Computer with Low-Level Parallelism, *Proc. of the 10th Int. Symp. on Computer Architecture*, pp. 151-157 (1983).
- 3) 元岡 達編: VLSI コンピュータ I, 岩波書店, 東京 (1984).
- 4) 富田眞治: VLSI とマイクロプログラミング, 情報処理学会計算機アーキテクチャ研資料, 57-2 (1985).
- 5) Bongiorno, V.: The CYBERPLUS Multiparallel Processor System, Donaldson, R. and Kreisler, M. N. (eds.), *Proc. of the Symp. on Recent Developments in Computing, Processor and Software Research for High-Energy Physics*, pp. 321-331 (1984).
- 6) Fisher, J. A.: Very Long Instruction Word Architectures and the ELI-512, *Proc. of the 10th Int. Symp. on Computer Architecture*, pp. 140-150 (1983).
- 7) 山口富士夫: コンピュータディスプレイによる図形処理工学, 日刊工業新聞社, 東京 (1981).
(昭和60年7月22日受付)
(昭和61年3月20日採録)



湯浅 真治 (正会員)

昭和36年生。昭和58年京都大学工学部情報工学科卒業。昭和60年同大学院修士課程修了。同年トヨタ自動車(株)に入社。現在、同社電算部第4技術電算課に勤務。



中田登志之 (正会員)

昭和32年生。昭和55年京都大学工学部情報工学科卒業。昭和60年同大学院博士課程修了。同年日本電気(株)入社。現在、同社C&Cシステム研究所に勤務。在学中は、LispマシンやQA-2の開発に従事。電子通信学会会員。



新實 治男 (正会員)

昭和31年生。昭和54年京都大学工学部情報工学科卒業。昭和56年同大学院修士課程修了。同年同大学工学部情報工学教室助手。現在に至る。計算機アーキテクチャ、図形処理システムなどに興味を持つ。電子通信学会会員。



富田 眞治 (正会員)

昭和20年生。昭和43年京都大学工学部電子工学科卒業。昭和48年同大学院博士課程修了。この間、零交さ波による音声合成の研究に従事。京都大学工学博士。同年京都大学工学部情報工学教室助手。昭和53年同助教授。現在に至る。計算機アーキテクチャ、並列処理システムなどに興味をもつ。著書(分担執筆)「計算機ハードウェア実験」「VLSI コンピュータ I」。電子通信学会, ACM, IEEE 各会員。ICOT・WG 委員。



萩原 宏 (正会員)

大正15年生。昭和25年京都大学工学部電気工学科卒業。NHKを経て、昭和32年京都大学工学部助教授。昭和36年同教授。現在に至る。工学博士。情報理論、パルス通信、電子計算機などの研究に従事。昭和31年度稲田賞受賞。昭和50年本学会論文賞受賞。昭和56~58年度本学会副会長。著書「電子計算機通論1~3」「マイクロプログラミング」など。電子通信学会, ACM, IEEE 各会員。