マルチメディア通信と分散処理 29－5
（１９８６．５．１５）

画像生成用ＳＩＭＤ型 マルチプロセッサシステム ＭＣ－２

平井 誠　日高 教行　浅原 重夫　鷺島 敬之
松下電器産業株式会社 無線研究所

筆者らが開発を進めてきた画像生成用マルチコンピュータシステムＭＣ－１の評価と検討を行い、ＭＣ－１の浮動小数点演算ユニットの稼働率は４３％にすぎず、稼働中も最大性能の１０％しか利用されていないことを明らかにする。この低い利用率を改善するため、レイトレーシング法における光線データの流れと物体データの流れとを別々に管理するユニットと、この２つのユニット間でＳＩＭＤ動作する多数のプロセッサエレメントを設け、空間分割アルゴリズムを適用することにより効率よくレイトレーシング画像を生成するシステムＭＣ－２を提案する。ＭＣ－２は大量の物体データの処理やアンチエイリアシング及び分散レイトレーシングの処理において特に有効である。

A SIMD Multiprocessor System MC－2 for Image Generation

Makoto HIRAI, Noriyuki HIDAKA, Sigeo ASAHARA, Takayuki SAGISHIMA
Wireless Research Laboratory, Matsushita Electric Ind. Co., Ltd.

By estimating the performance of an image generating system MC-1, which we have developed, it is showen that Arithmetic Processing Units of MC-1 operate during only 43% of the total processing time with the efficiency of 10% of its maximum performance. After considering the factors that limit the performance of MC-1, we propose an image generating system MC-2, which generates images about 100 times faster than MC-1, that is about 1GFLOPS. MC-2 is a SIMD multiprocessor system specialized for ray tracing space subdision algorithm. The MC-2 especially efficiently generates image of a scene that have many objects by distribute ray tracing.

1．はじめに

消費者主体型への消費動向の変化に従って、生産のあり方も少品種大量生産から多品種少量生産へと変化しつつある。こうした動きに応じて、髪型シミュレーションや着せ替えシミュレーション等にみられる消費者の好みに合せて製品を組合せて発注するための画像シミュレーションや、試作コスト削減のための画像シミュレーション等の必要性が高まっている。こうした画像シミュレーションに対する要求要件として以下の事項が挙げられる。

①実物どおりの感覚の画像を得られること。

このためにコンピュータグラフィックスの技術が必要であり、特にレイトレーシング法(1)が有効である。

②対話的に画像モデルを操作できること。

このために対話的な操作に耐え得る時間内に画像を生成できることと、モデリング及びシミュレーションの高い機能が必要である。

③実物を試作するよりも低コストであること。

画像生成システム単独であれば実物を試作する方が低コストであっても、統合的ＣＡＤ／ＣＡＭ環境として実現すれば全体として低コストに抑えられる。画像生成システムをＣＡＤ／ＣＡＭのネットワークにシミュレーション＆画像生成サーバとして接続する必要がある。

以上の要件を高品質化・高速化・低コスト化という視点から技術の現状と可能性について概観する。まず画像生成の高速化では、次の２方向からの研究成果が挙げられる。

①高速の演算器を並列に動作させる。(2)(3)(4)(5)(6)

②交差判定を効率よく行なうアルゴリズム。(2)(3)(5)(7)(8)

(2)のLINKS-1 システムでは物体データが少量の場合はプロセッサ数ｐに比例した処理時間が得られるが、物体データ量が各プロセッサの固有メモリ量を超えた場合はプロセッサ間通信のオーバーヘッドによってｐに対する処理速度の向上は飽和する。(5)では物体データを空間的に動的分割することで物体データ量がプロセッサの固有メモリ量を超えた場合でもＯ（ｐ¾）の処理速度が得られるとしている。(7)(8)では物体データを空間的に分割することにより、物体データ量に大きく依存しないような処理時間内でレイトレーシングアルゴリズムをシングルプロセッサで実行できるとしている。

次に高品質化では、次の２点が重要である。

①高解像度化。(9)

②アンチエイリアシング。分散レイトレーシング。(10)

(9)では2448×2048画素の解像度をもつカラーCRT に60Hz ノンインターレス表示するためのビデオプロセッサと400MHzのD/A コンバータの報告がある。(10)ではレイトレーシングの光線を従来のレイトレーシングの光線方向のまわりに多数発生させることで、アンチエイリアシング・物体の動きによって発生するブレを表現するモーションブラ・ふちのぼけた影の表示・ざらざらの表面に他の物体が写り込む様子の表示等のシミュレーションにより現実感の高い画像を生成できる。

低コスト化では次の２点を考える。

①ＶＬＳＩ化

メモリのＶＬＳＩ化によりメモリのビット当り単価は１～２年ごとに半額程度に落ちている。一方、演算回路についてもＶＬＳＩ化され、コンパクトで高いコストパーフォーマンスが得られるようになっている。今後の更なる高集積化を考えると、汎用品であるメモリのメモリ単独での高集積化によるメモリのコストダウンはある程度で飽和し、メモリと演算回路の組み合せによる特定のアプリケーション向きのＵＬＳＩ化の傾向が強くなると考えられる。ＵＬＳＩ時代のコストを考えるには、メモリや演算回路が半導体チップ上で占める面積で評価する必要がある。よって同じコストで高い処理性能を引き出すには、必要となるメモリ量を極小にし、多数の演算回路を並列に動作させ得るアルゴリズムが必要となる。

②ネットワーク化

対話的画像生成環境を実現するためには、ネットワーク中の各ワークステーションに高性能の画像生成システムを持つ必要がある。しかしワークステーションにおいて常に高品質の画像が必要であるわけではないことから、低コスト化のために高性能の画像生成システムは画像生成サーバとしてネットワークに接続し、対話的環境を維持するのに必要最小限の機能だけをワークステーションに残せばよいと考えられる。

本稿では、これまで筆者らが研究開発を行ってきた画像生成システムＭＣ－１を評価し、上記の視点からの見直しによって検討を行う。これに基いてSIMD型マルチプロセッサで画像生成を効率よく行うシステムＭＣ－２を提案する。

2．画像生成システムＭＣ－１の構成

３次元画像生成における物体データの座標値や表面属性は、浮動小数点データとして処理されるべきものであり、座標変換や隠面消去，物体表面の輝度計算等の演算は、加減乗除の四則演算を組み合わせて実行できる。

我々は上記の観点に立ち、高速の浮動小数点演算ユニット（ＡＰＵ）を開発し、このＡＰＵと16ビットマイクロコンピュータＭＣ68000 （ＭＰＵ）との対をユニットコンピュータ（ＭＣ）とし、ＭＣをＨＤＬＣ回線により結合したマルチコンピュータシステムＭＣ－１を開発した(3)。

図１はＭＣ－１のシステム構成図である。ＭＣ－１は、ルートコンピュータ（ＲＣ）とノードコンピュータ（ＮＣ）がＨＤＬＣ回線（４Ｍbps ）により図１の様に接続されたマルチコンピュータシステムであり、ＲＣによる監視のもとで、各ＮＣでは共通のプログラムが実行される。画像生成時、まずＲＣは、ＨＤＬＣ回線を用いてプログラム及びデータを一斉放送によりＮＣにダウンロードし、ＮＣ上のプログラムを起動する。その後ＲＣは、ブロック（スクリーンを細かく分割した矩形の領域）単位の負荷を各ＮＣに分散し、各ＮＣの実行を監視する。各ＮＣは、割り当てられたブロック内の各ピクセルの輝度を計算し、専用の高速イメージバスに出力する。各ＮＣからイメージバス上に出力された画像データは、データコレクタ（ＤＣ）により集められフレームメモリ（ＦＭ）に書き込まれる。

ＭＣのＡＰＵは、浮動小数点乗算器，加減算器，関数演算器（sin ，cos ，逆数，平方根）を持ち、それぞれ基本的に２ステージのパイプライン構成をとり、24ビットの浮動小数点演算を最大４Ｍ回／秒実行できる。また、プログラムメモリ及びデータメモリはそれぞれ２バンクからなり、ＡＰＵが演算を行っている間に、ＭＰＵは次の処理のためのプログラム及びデータを準備することができる。そのため、ＡＰＵへのプログラム及びデータの転送時間を無視することができ、ＡＰＵとＭＰＵは並行して動作する。

以下に、ＭＣ－１における画像生成手順を述べる。

ＲＣは、１画面の物体データを全てのＮＣにグローバル送信し、ＮＣの画像生成プログラムを起動する。ＮＣは物体及び光源を視点座標系に変換し、それぞれの物体を囲むbox を作成する。

次にＮＣは、ＲＣから割当てられた範囲のブロックに対し、各物体のbox を投影し各ブロックに写り込む可能性のある物体の番号を視点に近い順にソーティングしたリスト（ブロックリスト）を作成し、ＲＣに送信する。

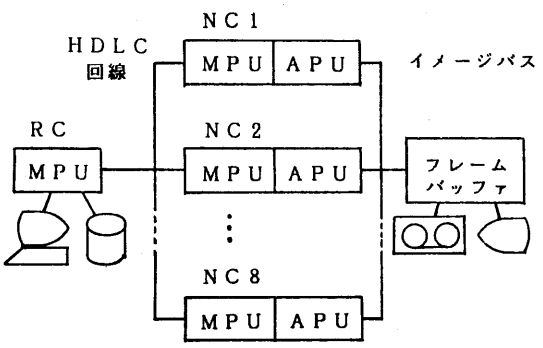


図１　ＭＣ－１のシステム構成

ＲＣは全てのＮＣからブロックリストを回収した後、それぞれのＮＣにブロック単位で負荷を割付け対応するブロックリストを送信し、以後ＮＣを監視しながら全てのブロックの割付けが終了するまでこれを行なう。

各ＮＣでは、割り付けられた各ブロックについて、隠面処理，影処理，輝度計算，反射・透過処理を行ない、画像データをイメージバスに出力する。

隠面処理では、z_buffer 法により、ブロックリスト中の物体データを処理し、ブロック内の各ピクセルに写り込む点の物体番号，ｚ値，法線ベクトルを求める。三角板はスキャンコンバージョンで、楕円体はbox 判定及び交差判定により写り込む点を求める。

影処理では、ブロック内の各ピクセルに写り込んだ点について、その点と光源を結ぶ直線と各物体の交差判定を行ない、光源からの光がさえぎられる場合にはその光源に対し影フラグをセットし、そのピクセルの輝度計算時にその光源を無視する。

輝度計算はPhong のモデルに基き、各ピクセルに写り込んだ点の座標，法線ベクトル，影フラグ，物体属性から輝度を計算しイメージバスに出力する。物体が反射・透過属性を持つピクセルは、対応する反射・透過フラグをセットし、画像データを出力しない。

反射・透過処理は、輝度計算後、反射・透過フラグがセットされているピクセルがある場合にのみ行なわれ、レイトレーシング法により各ピクセルの輝度を求める。その際、複数の光線の処理をまとめて行なうことにより高速化を図っている。

３．ＭＣ－１の評価と検討

写真１にＭＣ－１による画像生成例を示す。表１にこの画像を含め３種類の物体データに対するＭＣ－１とVAX-11/780との画像生成時間の比較を示す。これからＭＣ－１ではVAX-11/780（アクセレータなし）に比べて約

写真１　画像生成例　（玉と板）

５０倍の画像生成速度を実現していることがわかる。

一方、表２にVAX-11/780とＭＣ８台からなるＭＣ－１との命令実行速度及び浮動小数点演算速度の比較を示す。ＭＣ－１の命令実行速度はＭＰＵのMC68000 の命令実行速度１MIPSとＡＰＵの命令実行速度４MIPSを加え、ＭＣの台数８を掛けたものである。またＭＣ－１の浮動小数点演算速度はＡＰＵの浮動小数点加減算器，乗算器，関数演算器が並行してパイプライン動作をした場合のピーク値４×３MFLOPSにＭＣの台数８を掛けたものである。ＡＰＵの命令セットは低水準のマイクロプログラム的なものであり、VAX-11の命令実行速度と同等に扱えないが、一方ＡＰＵは１命令で多数の演算ユニットを同時制御できるため、逆にVAX-11の命令セットよりも高機能であると考えることもできる。表２で比較する限りでは、浮動小数点演算実行速度比よりも命令実行速度比によって、画像生成時間比が定まっているようにみえる。

そこでＡＰＵの各浮動小数点演算器の実行頻度を測定し、浮動小数点演算器の性能がどの程度まで生かされているのかを評価する。

まずＡＰＵの稼動率について評価する。ＡＰＵはＭＰＵの制御下で動作しており、ＭＰＵとＡＰＵ間でのデータ転送持ち合わせによってＡＰＵは停止状態となり、ＡＰＵの実効効率が低下する。写真１の画像を生成する際の各処理における全処理時間とＡＰＵの動作時間とを測定した結果を表３に示す。測定のためのタイマールーチンのオーバーヘッドにより全処理時間は表１の結果と正確に一致していないが、ＡＰＵは全画像生成時間の約４

表1　MC－1とVAX－11／780の画像生成時間

画　像	VAX－11／780	MC－1
ロ　ボ　ッ　ト	151’04”	3’03”（49．5倍）
球2つとチェッカーボード（写真1）	167’27”	4’36’（36．4倍）
球3つとチェッカーボード	318’38”	6’30”（49．0倍）

表2　MC－1とVAX－11／780の実行速度（ピーク値）

	VAX－11／780	MC－1（ピーク値）
命令実行速度	1MIPS	40MIPS（40倍）
浮動小数点演算速度	0．2MFLOPS	96MFLOPS（480倍）

表3　画像生成の各処理の時間とＡＰＵの動作時間
（写真１の画像生成時）

	全処理時間	ＡＰＵ動作時間	ＡＰＵ稼動率
隠面処理	9．28秒	1．49秒	16%
影付処理	71．16秒	25．39秒	36%
反射透過処理	217．64秒	112．72秒	52%
陰影処理	45．60秒	9．65秒	21%
計	343．68秒	149．25秒	43%

３％しか動作していないことがわかる。ＡＰＵの稼動率を考慮すれば、表１の実行速度比は表４のようになる。

次にＡＰＵの各浮動小数点演算器の実行頻度を求める。ＡＰＵの浮動小数点演算器の実行状態を直接モニタすることは容易でないため、ＡＰＵの動作をシミュレーションするＡＰＵシミュレーションプログラムを用いて測定した。写真１の画像の全画面の画像生成をシミュレートするためには膨大な時間がかかるため、反射屈折処理の比率の大きな部分と小さな部分と２ブロック（１ブロックは16×16画素）についてシミュレーションを行った。結果を表５に示す。２つのブロックの結果を単純に平均するとＡＰＵの実効浮動小数点演算速度は1.21MFLOPSとなりピーク値12MFLOPSの約10％の性能しか使われてないこととなる。ＡＰＵの実効浮動小数点演算速度を考慮すれば、表４の実行速度比は表６のようになる。

以上の性能評価をもとにＭＣ－１の性能改善について検討を行う。

①８台のＮＣの並列実行度

すべての物体データを各ＮＣが保持しており、画像生成実行中にＭＣ間で通信する内容は動的負荷分散のための画面ブロックの割合てのみであるため、並列実行のためのオーバーヘッドはほとんど無い。よってＮＣの台数に比例した画像生成速度が得られる。しかし、今回の実

表4　MC－1とVAX－11／780の実行速度比較
（APUの稼働率を43%とした）

	VAX－11／780	MC－1（APU稼動率43%）
命令実行速度	1MIPS	21.76MIPS（21.76倍）
浮動小数点演算速度	0.2MFLOPS	41.28MFLOPS（206倍）

表5　浮動小数点演算器使用頻度
（APUシミュレーション結果，APU動作時間に対する値）
（〔 〕内は非パイプライン動作の回数を示す）

演算器	反射屈折処理のあるブロック	反射屈折処理のないブロック	予想平均実効MFLOPS
加減数	1，583，850回 (0.87MFLOPS) 〔893，086回〕	119，967回 (0.56MFLOPS) 〔73，438回〕	0.72
乗算	747，381回 (0.41MFLOPS) 〔237，887回〕	91，609回 (0.43MFLOPS) 〔34，616回〕	0.42
関数	149，681回 (0.08MFLOPS) 〔146，384回〕	10，485回 (0.05MFLOPS) 〔10，425回〕	0.07
APU動作時間	1.817秒	0.213秒	合　計 1.21
非パイプライン率	51.5%	53.4%	平　均 52.4%

表6　MC－1とVAX－11／780の実効速度比較
（APUの稼動率43%，1.21MFLOPS）

	VAX－11／780	MC－1
命令実行速度	1MIPS	21.76MIPS（21.76倍）
実効浮動小数点演算速度	0.2MFLOPS	4.16MFLOPS（20.8倍）

験では物体数が少ないため、すべての物体データを各ＮＣが保持することができたが、実際に画像生成を実用に供する場合は物体データが多量になるため、各ＮＣは物体データの一部分を保持し、必要に応じてＲＣより物体データを転送してもらうことになる。この場合，物体データ転送処理時間と画像生成時間との比のＮＣ台数以上にＮＣを増設しても画像生成時間は短縮できなくなる。

この解決方法として、ＲＣ－ＮＣの星状結合を階層化することにより、各星状結合内でのＮＣ台数を適切な数に抑えつつ、全体のＮＣ台数を増設する方法が報告されている(2)。我々は別の解決策として、ＮＣを増設するかわりにＮＣ内部の演算プロセッサを並列化することにより、プロセッサの増設を図る。

②ＡＰＵの稼動率

ＭＰＵとＡＰＵの役割分担として、ＭＰＵは16MBの大きな線形アドレス空間と種々のアドレッシングモードとを持ち、複雑で大量の物体データを管理し、画像生成を高速化するためのアルゴリズムとデータ構造とを扱うのに適しており、ＡＰＵは４Ｋワードの限られたデータメモリと、アドレスレジスタへのメモリアドレスの設定及びインクリメントだけというアドレッシングモードのため、単純なデータ構造のデータを高速にパイプライン演算するのに適している。ＭＣ－１では以上のように役割を専用化し、ＭＰＵとＡＰＵの協調によって処理速度を向上させている。

しかし、ＡＰＵの稼動率が全体で43%とあまり上がっていないのは、表３からわかるように各処理によってＭＰＵとＡＰＵとの負荷のバランスがとれていないためである。この原因として以下のものが考えられる。

1. ＡＰＵでの処理に必要であると予想されるデータをＭＰＵが準備するのに多くの手間がかかるのに反して、本当にＡＰＵでの処理に必要なデータは一部分であり、ＡＰＵの処理量は少量である場合がある。
2. ＡＰＵのデータメモリが４Ｋワードしかないため、多くの種類のデータをデータメモリに詰め込んで、複雑な構造のデータをＡＰＵに転送する場合には、ＡＰＵの整数ユニットでのアドレス計算の負荷が大きくなる。特に反射透過処理の場合。
3. ＡＰＵでの演算が簡単な割に、ＭＰＵでのデータのソート処理及びリスト処理の負荷が大きい場合がある。

1.についてはＡＰＵの処理の流れに合わせてＭＰＵがデータを準備できればよい。2.についてはＡＰＵのアドレス演算能力を上げてやればよい。もしくは、ＡＰＵの処理単位を16×16画素より小さくし、１度にＡＰＵへ転送するデータ量を小さくすることで、データメモリに複雑な構造のデータを入れないようにすれば、複雑なアドレス計算をＡＰＵで行うことを避けられる。3.については画像生成アルゴリズムの問題であり、アルゴリズムを改善するか、ＭＰＵの性能を上げるかしかない。ＭＣではＡＰＵの性能に対しＭＰＵの性能が低く、バランスがとれていないといえる。

③ＡＰＵの実効浮動小数点演算速度

ＭＣ－１ではＡＰＵ最大性能の12MFLOPSの約10%しか使用されていない。これは３本のパイプライン浮動小数点演算器を同時に使用することはほとんどないことを示している。また表５に示したようにパイプライン演算器を使用したうちの52.4%はパイプライン動作させていないことがわかる。さらに浮動小数点演算器を全く用いないで、アドレス計算や条件判断等の処理を行なっている命令サイクルもかなりの割合を占めていると考えられる。表７に写真１の画像の反射透過処理のあるブロック(16×16画素）の画像を生成する時のＡＰＵの浮動小数点演算以外の命令のうちのある一部分の命令の実行頻度をＡＰＵシミュレータを用いて測定した結果を示す。これらはＡＰＵの同一の命令フィールドを占めるため同時には実行できない。つまり全ＡＰＵ実行時間の61%でこれらの命令が実行されており、命令の性格上浮動小数点演算と並列して実行されるものは少ないことから、実行浮動小数点演算速度を低下させている主原因と考えられる。その理由として、いくつかのまとまった浮動小数点演算を実行する前にデータメモリのアドレスを示すアドレスレジスタの初期化やアキュムレータや定数レジスタとして用いる浮動小数点レジスタの初期化等を行い、最後に結果の値を判断するというようなプログラム構成が基本になっているということが挙げられる。

これを改善するには、スーパーコンピュータでいうところのベクトル長を長くする、つまりできるだけ多くのまとまった浮動小数点演算を実行することが必要である。ところが、画像生成においては高々３次か４次の行列・ベクトル計算しかなくベクトル長を長くとれないとされている(2)。

しかし、上記の考え方は１本の光線に対して各物体との交差判定をし、輝度を計算し、反射透過処理を行なう

表７　ＡＰＵの命令実行頻度のシミュレーション結果
（透過反射処理のあるブロック（１６×１６ドット））

ＡＰＵの命令	実行回数	実行頻度
整数データ転送	1，104，524回	16.6%
整数イミーディエイト	1，107，580回	16.7%
浮動小数点イミーディエイト	235，238回	3.5%
条件判断	1，128，281回	17.0%
無条件ジャンプ	126，108回	1.9%
サブルーチン呼出	146，507回	2.2%
サブルーチン復帰	146，507回	2.2%
	計	61.0%

場合のものであり、⑹にあるようにいくつかのまとまった本数の光線に対して上記の処理を行なえばベクトル長を長くとることができる。この方法は、画像を高品質化する場合、すなわち画面の画素数を増加させ⑼、アンチエイリアシングや分散レイトレーシング⑽を行う場合、光線の本数をますます増加させる必要があるため、さらに有利になると考えられる。

4. ＭＣ－２の構成

以上の考察に基づいて、画像生成用ＳＩＭＤ型マルチプロセッサシステムＭＣ－２を構成する。まずＳＩＭＤに適したレイトレーシングアルゴリズムについて述べ、それに基づいてハードウェアを構成する。

4.1 ＳＩＭＤ向きレイトレーシング法

① まず画像を生成する前に物体全体が収まる箱を作成し、この箱を分割してvoxel を作成する⑻。各voxel に包まれる物体の基本形状数がある数以下になるまでvoxel の再分割を繰返す。

② 各voxel ごとに、視点から表示スクリーンを通って、そのvoxel へ入射する光線データを各voxel ごとの光線キューに入れておく。

③ 一定のベクトル長に達した光線キューに入っている光線データと、そのvoxel 内に存在する基本形状との最も視点に近い交点を求める。交点がなければ、その光線の進行方向に隣接するvoxel の光線キューに入れなおす。進行方向にvoxel がなくなれば、その光線データは消去する。

④ 交点の得られた光線については、交点における輝度を求め、その光線の輝度の画素の輝度に対する寄与率を掛けて画素の輝度に加える。交点における表面の属性に従って反射光線，透過光線，光源へ向かう光線を生成し、そのvoxel の光線キューに追加する。

⑤すべての光線キューが空になるまで、③④を繰り返す。

以上の処理において、アンチエイリアシング処理が必要な場合には、次のような処理を加える。スクリーン上の隣接する画素間で、ある一定値以上の輝度差があるときには、画素に対して数本の光線を出し輝度を再計算する。これらの光線のその画素に対する寄与率の合計は１となる。モーションブラや半影等を分散レイトレーシングで実現する場合も同様である。

このアルゴリズムがＳＩＭＤ型マルチプロセッサ向きであるとするのは以下の理由からである。

①一定の長さのベクトル長が確保できる。

②voxel というローカルな空間内での処理単位に分割するため、データのコヒーレンス性がある程度保て、処理の流れがデータに強く依存しても、処理効率の低下を抑えられる。

4.2 ＭＣ－２のハードウェア構成要件

まず上記のＳＩＭＤ向きレイトレーシングアルゴリズムを処理するために必要となるハードウェア構成要件を以下に列挙する。

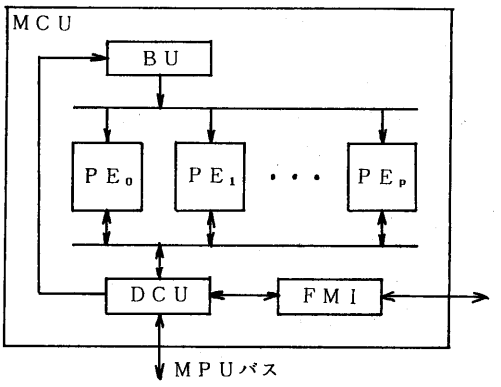
①物体データvoxel 単位で管理し、マルチプロセッサに対して物体データ放送するユニットＢＵ
(BU:Broadcasting Unit)

②光線データをvoxel 単位で管理し、マルチプロセッサに対して光線データを分配・回収するユニットＤＣＵ
(DCU: Distributing & Collecting Unit)

③輝度データをpixel 単位で管理し、マルチプロセッサからの輝度データを累積するユニットＦＭ

④ＤＣＵからの光線データに対し、ＢＵからの基本形状データとの交差判定を行い、輝度の計算、新たな光線データの生成等を行うプロセッサエレメントＰＥ

以上の構成要件の全体の管理は、アルゴリズムの流れからわかるようにＤＣＵで行なう。またＰＥの処理の流れは物体データに依存するため、ＢＵからすべてのＰＥに対して実行制御命令を放送する。以上の構成を図２に示す。このＭＣ－２の基本構成要素をモデリング＆コンピュートユニット（ＭＣＵ）と呼ぶ。



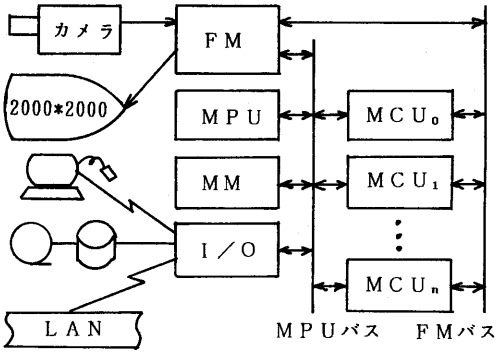
ＭＣＵ：Modeling ＆Computing Unit
ＢＵ：Broadcasting Unit
ＰＥ：Processor Element
ＤＣＵ：Distribing & Collecting Unit
ＦＭＩ：Frame Memory Interface

図２ ＭＣ－２の基本構成要素ＭＣＵ

4.3　MC－2の実現構想

図3にシステム構成を示す。MC－2はネットワークシステム中の1ノードとして存在し、ネットワークシステムの動的負荷分散機能により、多数のMCUをもつノードは、モデリング&コンピュートサーバとしての性格を強くもち、MCUの少ないノードはワークステーションとしての性格が強くなる。

図4にMCUの構成を示す。BU，DCUは各々8MIPSの性能をもち、PEは加算器8MFLOPS，乗算器8MFLOPS，計16MFLOPSの性能をもつ。プログラムメモリ及びデータメモリは1MB 単位で増設できる。PEのレジスタははは4Kワードあり、サイクルスチールで3ポートレジスタとしている。2ポートはPE用，1ポートはDCU用である。レジスタアドレスコントローラはサイクルスチールの制御とDCU－PE間のデータ交換を効率よく行なうためのベースアドレス修飾とを行う。フレームメモリインターフェースはフレームメモリをDCUのアドレス空間内にマッピングし、複数のMCU間の競合を調停する。またDDA(Degital Differencial Analysis) を用いた描画ハードウェアを用いてスキャンラインアルゴ



FM：Frame　Memory
MPU：Main Processing Unit
MM：Main Memory
MCU：Modeling &Computing　Unit
LAN：Local　Area Network

図3　MC－2システム構成

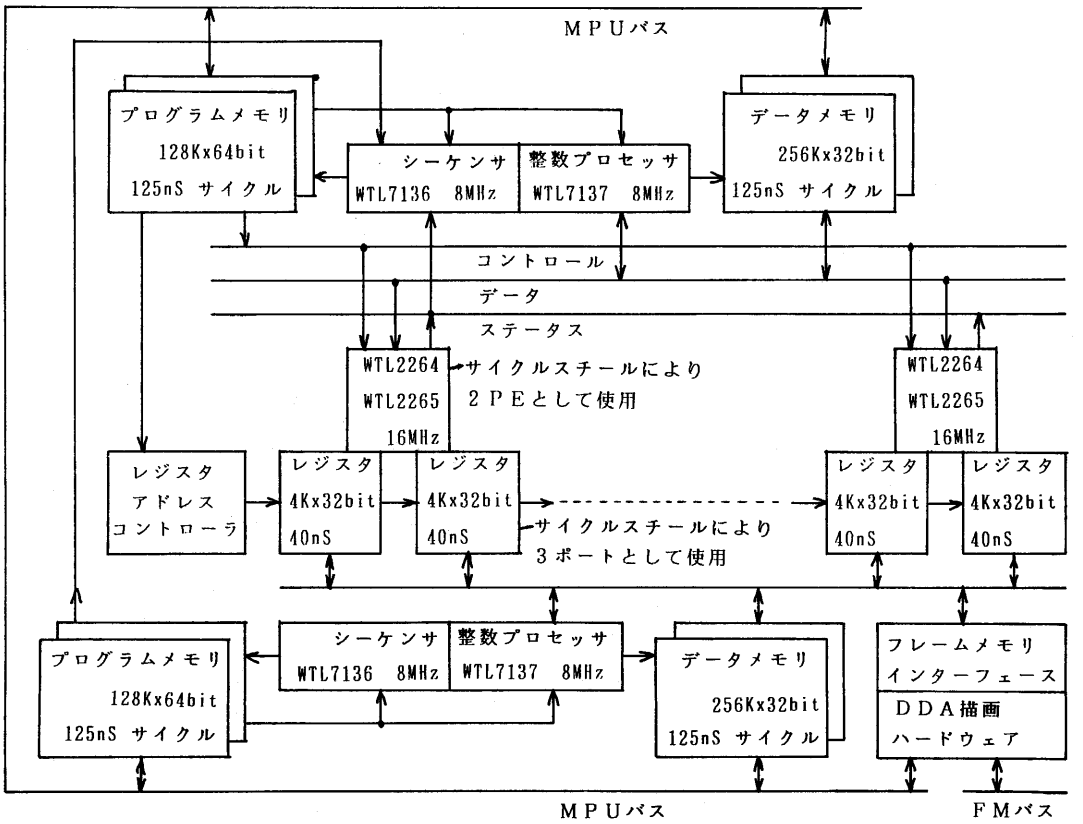


図4　MCUの構成

リズムによる画像の生成を効率よく行うこともできる。

ＰＥボードには、WTL2264,WTL265が各２個ずつ載り、１ボードで４ＰＥとなり最大64MFLOPSの性能となる。８枚のＰＥボードから成るＭＣＵを２セット持つＭＣ－２はＭＣ－１と同一サイズのラックに収まり、最大１GFLOPSの性能を発揮できる。

4.4　ＭＣ－２の構成についての考察

ＭＣ－１に対する構成上の利点について以下に述べる。

①ＭＰＵ－ＭＣＵ－ＰＥという階層化と各階層間の転送速度の改善によりプロセッサ数をＭＣ－１よりかなり多くしても処理速度が飽和しない。

②ＭＣＵをＳＩＭＤ化することによりプログラムメモリ及びデータメモリを共有化し、演算回路に対するメモリ回路の比率を小さくでき、ＵＬＳＩ化へと一歩前進した。

③ＭＣ－１におけるＭＰＵとＡＰＵの役割分担を明確化することでＤＣＵとＢＵという構成とし、ＭＣ－１でみられた種々のＭＰＵ－ＡＰＵ間のオーバーヘッドを解消し得る。

④アンチエリアシングや分散レイトレーシング等の高品質の画像生成からスキャンライン法等の低品質の画像生成にわたる広範囲の画像生成を品質にみあった時間で生成できる上、これらの画像の合成をＦＭ上で行える。

⑤ＭＣ－１に比べてデータメモリ量，プログラムメモリ量を大幅に増設可能とした。またＰＥ－ＤＣＵ間での同期・データ転送を簡単化したことで、ＰＥ－ＤＣＵ間でのデータパケットを小容量化し、パケット流量を上げることができる。よって、きめの細かいアルゴリズムの実現を容易にした。

⑥ＭＣ－１とほぼ同程度の匡体サイズに、最大浮動小数点演算速度でＭＣ－１の96MFLOPSに対し１GFLOPSと１桁性能向上した。レイトレーシングによる画像生成では２桁程度の性能向上が期待でき、写真１の画像を数秒で生成する。また複雑なシーンの画像生成においても、空間分割アルゴリズムの特性上[8]、ほぼ同程度の時間で生成できる。

5.　まとめ

ＭＣ－１の評価と検討により画像生成用マルチコンピュータシステムの実現上の種々の問題点を明確化し、ＭＣ－２の基本構成について述べた。

ＭＣ－２はレイトレーシング法による画像生成の高速化をめざしたＳＩＭＤ型マルチプロセッサシステムである。その構成上の特徴は物体データの流れと光線データの流れとを別々に管理するユニットを設け、物体データの空間分割アルゴリズムを採用することにより、この２つのユニットの間で多数のプロセッサエレメントを効率よく動作させることである。ＭＣ－１と同一サイズのハードウェア構成で約２桁の処理速度向上を達成する見通しが得られた。

今後、ＭＣ－２の実現を行いつつ、ＳＩＭＤ向きレイトレーシング法のシミュレーション及び評価を行う。そして、ネットワークシステム化とワークステーションとしてのマンマシンインターフェースについても検討を進めてゆく予定である。

謝辞　日頃御指導賜わる無線研究所　中島昌也所長，大阪大学工学部　大村皓一助教授，白川　功助教授，イリノイ大学　出口　弘氏に感謝する。

参考文献

〔１〕T.Whitted:"An Improved Illmination Model for Shaded Display"CACM,Vol.23,NO.6,pp343-349 (1980-6)

〔２〕出口他：”視線探索法による画像生成のための木構造並列処理システム”，信学論Vol.J69-D,NO.2 (1986-2)

〔３〕日高他：”マルチコンピュータ画像生成システムMC-1”，情処計算機アーキテクチャ研資85-CA-58-5(1985-6)

〔４〕吉田他：”グラフィックス計算機SIGHT の基本構成”，情処計算機アーキテクチャ研資85-CA-60-4 (1985-12)

〔５〕M.Dippe et al:"An Adaptive Subdivision Algorithm and Parallel Architecture for Realistic Image Synthesis",ACM,SIGGRAPH'84, Computer Graphics,18-3,PP 149-158(1984-7)

〔６〕D.J.Plunkett et al:"The Vectorization of a Ray-Tracing Algorithm for Improved Execution Speed",IEEE CG&A pp52-60(1985-8)

〔７〕山本他：”変形オクトリーを用いた光線追跡の高速化アルゴリム”，情処グラフィクスとＣＡＤシンポジウム資料PP45-51(1985-12)

〔８〕村上："Voxel分割を用いた高速化レイトレーシング”，情処第32回全国大会資料5V-2,PP2082, (1986-4)

〔９〕辻岡他：”超高精細高速曲面表示技術の開発”，情処グラフィクスとＣＡＤシポジウム資料，PP167-175(1985-12)

〔１０〕Cook et al:"Distributed ray tracing", SIGGRAPH'84 Compter Graphics 18-3,pp137-145 (1984-7)