



Title	3次元画像生成のためのマルチコンピュータシステムに関する研究
Author(s)	河合, 利幸
Citation	大阪大学, 1989, 博士論文
Version Type	VoR
URL	https://hdl.handle.net/11094/1656
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

3次元画像生成のための
マルチコンピュータシステムに関する研究

1989年 1 月

河 合 利 幸

内容梗概

本研究は大阪大学大学院工学研究科後期課程（電子工学専攻）在学中に寺田研究室において行った研究のうち、3次元画像生成のためのマルチコンピュータシステムに関する研究をまとめたものである。

第1章序論においては、本研究の目的並びにその工学上の意義、及びこの分野での研究の現状について述べ、本研究で得られた新しい成果について概説する。

第2章においては、まず我々が設計・試作した、視線探索法を用いた並列画像生成システムLINKS-1の概要と、実験・運用により明らかになった諸問題について述べる。また、並列処理の効率を上げるためには、演算処理速度の向上と通信処理能力の強化が必要となることも示す。

第3章においては、演算処理を高速化するためにはユニットコンピュータ内部におけるデータ転送の高速化が重要であることに着目し、新たに設計した、並列画像生成システムLINKS-2を構成するユニットコンピュータの設計方針とその概要について述べる。また、その過程で用いられた乗算器の自動構成手法についても述べる。

第4章においては、ユニットコンピュータ内部でのデータ転送の高速化手法、並びにユニットコンピュータの動作について述べる。また、シミュレーションにより、その演算処理能力を明らかにする。その結果として、データメモリのインタリーブ構成を始めとする2項演算の並列パイプライン処理が画像生成処理の高速化に有効であることを示す。

第5章においては、並列処理システムを構成するために必要なユニットコンピュータ間の通信方式について述べ、共有バスを用いた高速ブロック転送方式が有効であることを示す。また、並列処理システムの構成例についてもふれる。さらに、本方式により、画像生成速度、並列処理時の効率共に改善されたことをシミュレーションにより明らかにする。

第6章においては、空間分割法を視線探索法の並列処理化に適用する手法について述べる。本論文では、空間を角錐状に分割し、フレーム間コヒーレ

ンスを利用した負荷分散を行う。この結果、1プロセッサあたりの必要メモリ量は画面分割法（バッチ転送方式）に比べて大幅に削減でき、並列処理効率は、ほぼプロセッサ数に比例することが確認できた。

第7章においては、本研究で得られた結果と残された課題についてまとめている。

関連発表論文

論文誌等に発表した論文・レター（査読付き）

- (1) 河合利幸, 山下伸一, 大野廣司, 吉村浩, 西村仁志, 下條真司,
宮原秀夫, 大村皓一,
”並列画像生成システムLINKS-2のアーキテクチャ”,
情報処理学会論文誌, Vol.29, No.8, pp.729-740, 1988,8.
- (2) 河合利幸, 藤田卓志, 大村皓一,
”2重節点をもつスプライン基底の一構成法”,
電子情報通信学会論文誌(D), Vol.J71-D, No.6, pp.1149-1150, 1988,6.
- (3) 西村仁志, 平井誠, 河合利幸, 河田亨, 白川功, 大村皓一,
”分布関数による物体モデリングと画像生成の一手法”,
電子通信学会論文誌(D), Vol.J68-D, No.4, pp.718-725, 1985,4.

国際会議等に発表した論文

- (1) Hitoshi NISHIMURA, Hiroshi DEGUCHI, Toshiyuki KAWAI,
Shinichi YAMASHITA, Toru KAWATA, Isao SHIRAKAWA, Koichi OMURA,
”A multicomputer system LINKS: Its architecture and graphics
applications”, Proceedings 1984 International Symposium on
Industrial Electronics, Control and Instrumentation, Vol.1,
pp.270-274, Oct. 1984.

- (2) Hitoshi NISHIMURA, Hiroshi DEGUCHI, Toshiyuki KAWAI,
Shinichi YAMASHITA, Toru KAWATA, Isao SHIRAKAWA, Koichi OMURA,
"A parallel processing scheme for 3D image generation and its
applications", Proceedings China 1985 International Conference on
Circuits and Systems, pp.280-283, June 1985.

研究会等に発表した論文

- (1) 出口弘, 河合利幸, 中西隆, 西村仁志, 河田亨, 白川功, 大村皓一,
"プログラミングシステム LINKS-C",
情報処理学会マイクロコンピュータ研究会資料, 24-3, 1982, 11.
- (2) 河合利幸, 吉村浩, 出口弘, 西村仁志, 河田亨, 白川功, 大村皓一,
"画像データ操作システム LINKS-DMS",
情報処理学会マイクロコンピュータ研究会資料, 24-4, 1982, 11.
- (3) 西村仁志, 吉村浩, 出口弘, 辰己敏一, 河合利幸, 河田亨, 白川功,
大村皓一, 尾崎弘,
"マルチマイクロコンピュータにおける画像生成のための画面分割方式",
電子通信学会技術研究報告, CAS82-144, pp.31-36, 1983, 2.
- (4) 河合利幸, 西村仁志, 出口弘, 河田亨, 白川功, 大村皓一,
"LINKS-1によるアニメーションの製作について",
情報処理学会グラフィックスとCAD研究会資料, 14-4, 1984, 9.
- (5) 木本泰治, 河合利幸, 那須雅樹, 河田亨, 白川功, 大村皓一,
"コンピュータグラフィックス用画像データベースシステム",
情報処理学会コンピュータビジョン研究会資料, 38-3, 1985, 9.

- (6) 河合利幸, 若井裕久, 正田博司, 西村仁志, 近藤仁志, 高山浩一郎,
出口弘, 白川功, 大村皓一,
"画像生成用マルチコンピュータシステムとプロセッサについて",
情報処理学会グラフィックスとCAD研究会資料, 19-4, 1985, 10.
- (7) 河合利幸, 木本泰治, 那須雅樹, 河田亨, 白川功, 大村皓一,
"コンピュータグラフィックス用画像データベースシステム",
NICOGRAPH'85論文集, pp.21-29, 1985, 11.
- (8) 河合利幸, 大村皓一,
"LINKS-2",
NICOGRAPH'86論文集, 1986, 11.
- (9) 栗山繁, 河合利幸, 大村皓一,
"カーディナルスプラインを用いた自由曲面のモデリング手法",
情報処理学会グラフィックスとCAD研究会資料, 25-3, 1987, 2.
- (10) 藤田卓志, 岸良行, 栗山繁, 河合利幸, 大村皓一,
"多重節点を持つスプライン基底関数を用いた曲面補間法",
テレビジョン学会技術報告, ICS88-16, IPA88-11, 1988, 2.

全国大会等に発表した論文

- (1) 山名岳志, 西村仁志, 河合利幸, 河田亨, 白川功, 大村皓一,
"メタボール技法による自由曲面の創成と高速画像生成法",
電気関係学会関西支部連合大会, G14-7, 1984, 11.

(2) 河合利幸, 大村皓一,

”映像の言語的表現とコンピュータグラフィックス”,
電気関係学会関西支部連合大会, S12-8, 1986, 11.

(3) 大村皓一, 河合利幸,

”グラフィックス専用マシンLINKS-1, 2”,
第17回画像工学コンファレンス, 12-1, 1986, 12.

3次元画像生成のためのマルチコンピュータシステムに関する研究

目 次

第1章 序 論	1
---------------	---

第2章 LINKS-1の概要と問題点

2. 1 緒言	5
2. 2 視線探索法	5
2. 3 形状定義手法	8
2. 4 LINKS-1のシステム構成	10
2. 5 LINKS-1における並列処理方式	11
2. 6 LINKS-1における問題点	12
2. 7 結言	15

第3章 LINKS-2のユニットコンピュータ

3. 1 緒言	17
3. 2 ユニットコンピュータの設計方針	17
3. 3 ユニットコンピュータの命令	18
3. 3. 1 データ型	18
3. 3. 2 命令形式	19
3. 3. 3 アドレッシングモード	20
3. 4 ユニットコンピュータの構成	23
3. 4. 1 データプロセッサ	25
3. 4. 2 インデックスユニット	28
3. 4. 3 クロスバースイッチとデータメモリ	29
3. 4. 4 シーケンサユニットとプログラムメモリ	29
3. 4. 5 チャネルプロセッサ	29
3. 5 乗算回路の自動構成手法	30
3. 6 結言	32

第4章 演算処理方式とその評価	
4. 1 緒言	3 5
4. 2 データの配置	3 5
4. 3 クロスバースイッチの制御	3 6
4. 4 UCの動作	3 7
4. 5 演算処理方式の評価	3 9
4. 6 結言	4 2
 第5章 ユニットコンピュータ間通信方式とその評価	
5. 1 緒言	4 3
5. 2 UC間通信方式の特徴	4 3
5. 3 チャンネルプロセッサの構成と機能	4 4
5. 4 LCBの状態遷移とUC間通信手順	4 6
5. 5 並列処理システムの構成	4 9
5. 6 通信処理方式の評価	5 1
5. 7 結言	5 6
 第6章 空間分割法による並列画像生成	
6. 1 緒言	5 7
6. 2 並列化手法	5 9
6. 3 空間分割手法	6 0
6. 4 負荷分散手法	6 3
6. 5 シミュレーションによる評価	6 5
6. 6 結言	7 1
 第7章 結 論	7 3
 謝 辞	7 5
 参考文献	7 8

第 1 章 序 論

コンピュータグラフィックス(CG)は、計算機内に物体等のモデルを構築することにより、その形状、色彩、質感、運動等を視覚化する技術の総称である。CGは、1963年のSutherlandによるスケッチパッド(Sketchpad)に端を発すると言われている[SUTH63]。その応用分野は今や、機械設計、建築・土木設計、分子設計、TV放送・映画、芸術、視聴覚教育、フライトシミュレータ等、広い範囲に及び、急速に拡大しつつある。

CGにおいて達成しなければならない第一義的な技術目標は、応用分野によって異なる。例えば、フライトシミュレータでは、実時間応答性が最優先される[SCHA81][YAN85]のに対し、芸術分野では映像の品質・写実性が重要視される。しかし、フライトシミュレータにおいてもその映像はリアルであることが望ましく、芸術分野においても画像生成時間は短い方が望ましい。このように、よりリアルな画像をより高速に生成することこそ、各応用分野に共通する課題であり、CGにおける根元的な課題の一つである。

この課題に対して、最近では、映像化アルゴリズムに内在する並列処理の可能性に着目した専用マシンの研究・開発が各所で行われるようになってきた[NIIM85][YOSD88]。

これらのマシンのアーキテクチャは、使用する映像化アルゴリズムを反映したものとなっており、実用的な計算速度を重視したアルゴリズムであるスキャンライン法[WYLI67][BOUK70][BLIN80]、Zバッファ法[NEWE72][CATM75]に基づくものと、写実性を追求したアルゴリズムである視線探索法[WHIT80][OKAM81]に基づくものの2つに分類することができる。前者に属するものとしては、EXPERTS[NIIM84][OHTA85]、Geometry Engine[CLAR82]、G-PSYCO[KUBO80,83,84]、Pixel-Planes[FUCH85]等があり、これらの多くはパイプライン構成を採っている。後者には、CAP[SATO85a,b,c][MURA86][ISHI88]、MC-1/2[HIDA85][NAKS85][HIRA86]、SIGHT[YOSD85][TAKA87]、MAGIC[TAMA87,88]等、マルチプロセッサ構成を採用したものが多い。

我々は、写実性重視の立場から、反射・屈折等の表現が容易であり、高品質な陰影画像を得ることができる視線探索法を用いた並列画像生成システム L I N K S - 1 を他のシステムに先駆けて試作した[DEGU82][DEGU84a,b][DEGU86][HIRA84][KAWT83][NAKA82][NISH82][NISH83c][NISH84][NISH85c][YAMA83][YOSH82]。L I N K S - 1 は、CGによる動画制作を主目的としており[KAWA84]、本格的な画像生成専用システム開発のための実験システムとして設計されている。広義にはL I N K S - 1 は、モデリングシステム[KAWA82]、データベースシステム[SHIM85][KAWA85b]、アニメーションシステム、画像生成システムの4つのサブシステムからなる。1982年10月の試作完成以来、数多くの動画制作[FUKU84][KAWG83,84,85][MIYA85][NAKG85]を行いつつ実験を繰り返してきた[FUNA85][NISD84][NISH83a,b][NISH85b]。

これらの多数の実験結果から、算術演算速度およびユニットコンピュータ(U C)間の通信処理速度の不足が、システム全体の画像生成処理速度向上の妨げになっていることがわかってきた。そこでこれらの結果を基に、新たな並列画像生成システムL I N K S - 2 の設計・製作を開始した[KAWA85a][KAWA86a][OMUR86]。

L I N K S - 2 は、視線探索法による実用的な対話型CGシステム構築に必要な処理能力の達成を目標として設計された。L I N K S - 2 のU C は、L I N K S - 1 のU C と比較して、演算能力と通信機能が大幅に強化されている。本U C は、32ビット浮動小数点演算を並列処理するデータプロセッサ、アドレス計算を行うインデックスユニット、通信を制御するチャネルプロセッサ等主要ユニットが非同期に並列パイプライン動作する。メモリアクセス競合を削減し、データ転送を並列化するため、各ユニットはクロスバースイッチにより、4ウェイにインタリーブされたデータメモリと結合されている。これら主要部分は、カスタムV L S I を用いて実現した。その結果、本U C 1 台の画像生成速度はL I N K S - 1 の約50倍に達する。また、並列処理効率についても改善が認められた[KAWA88b]。

L I N K S - 2 の開発と並行して、空間分割法による視線探索法の並列化を検討し、シミュレーションにより評価を行った。

従来行われてきた並列化は、視線探索法が画素ごとに全く独立に計算可能であることを利用した画面分割法[DEGU84b][NISH83a][YOSH82]によるものである。この方式では、各プロセッサは基本的に3次元空間を構築する全物体データを必要とする。このため、メモリの使用効率が悪化する、あるいは全物体データを持つ特定のプロセッサに通信が集中しプロセッサ数に見合う性能が得られなくなる等の問題点がある[FUNA85][NISH85b][DEGU86]。

これらの問題点を解決するためには、物体データやプロセッサ間通信をうまく分散させる並列化手法が必要である。そのような並列化手法の1つに空間分割法[CLEA83][DIPP84]がある。空間分割法は、計算機内に構築された3次元空間をいくつかの部分空間に分割し、視線を各部分空間に進行させながら、それぞれの部分空間で視線探索法を実行する方式である。しかし、空間の分割方式や、各プロセッサにおける負荷の均一化等の問題が新たに生じる。

本論文では、空間を角錐状に分割し、フレーム間コヒーレンスを利用した負荷分散を行う。このような手法により、単純でしかも効率的な並列処理が可能となる。この結果、1プロセッサあたりの必要メモリ量は、画面分割法（バッチ転送方式）に比べて大幅に削減でき、並列処理効率に関しては、ほぼプロセッサ数に比例することが確認できた。

第2章 LINKS-1の

概要と問題点

2. 1 緒言

LINKS-1は、マルチマイクロプロセッサ構成のユニットコンピュータを、星状ないしは木構造状に接続した並列画像生成システムであり[NAKA82][NISH83c][DEGU86]、主要言語としてC言語[KERN78][DEGU82]を用いている。現在までに、動画作成のためのマンマシンインタフェース[KAWA82,84,85b][SHIM85]・新しい形状表現モデル[HIRA83][YAMN84][NISH85a][KURI87][FUJI88][KAWA88a]等の開発、各種シミュレーション実験および処理方式の改良・高速化を行ってきた[YAMA83][NISD84][NISH83a,b][NISH85b,c][FUNA85][DEGU86]。

本章では、画像生成手法として用いた視線探索法[WHIT80][OKAM81][DEGU84b]の特徴、形状定義手法、LINKS-1の構成および並列処理方式について概説する。さらに、実験および運用により明らかになった諸問題について述べ、新たなシステムの必要性を論じる。

2. 2 視線探索法

リアルな3次元画像生成を行うには、計算機内に構築した物体モデルのうち隠れて見えない面を除き、見える面を選び出し（隠面消去[SUTH74]）、その面上の各点の明るさを決定する（輝度計算）必要がある。

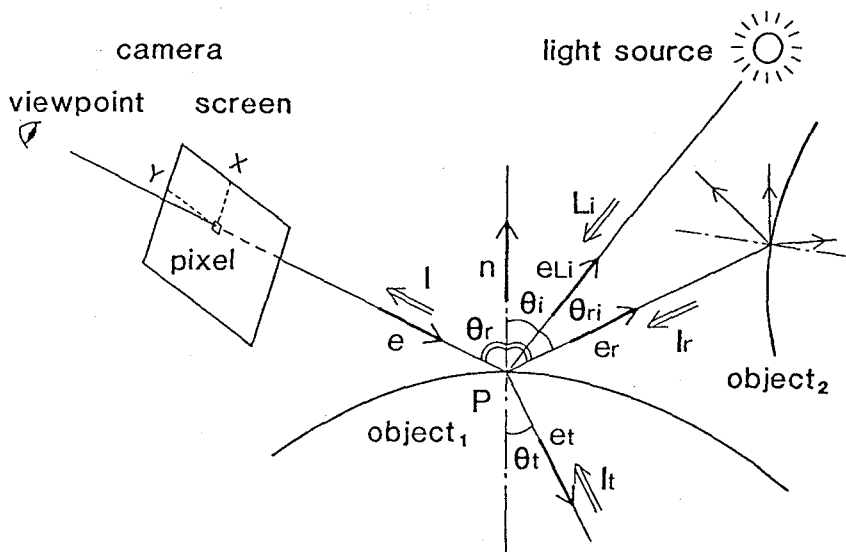
代表的な隠面消去法に、Zバッファ法[NEWE72][CATM75]、スキャンライン法[WYLI67][BOUK70][BLIN80]、エリアコヒーレンス法[WARN69]、優先順位法[BENN75][FUCH80]等がある。一般に隠面消去を高速に処理するためには、物体またはその像のもつ局所的に類似な性質（コヒーレンス）を利用する。これらのアルゴリズムは、物体モデルとして多面体近似を基本としており、コヒーレンスを利用することで比較的高速に処理できる。また、実時間応答性が重視されるフライトシミュレータ[SCHA81][YAN85]等ではハードウェア化も行われている。

一方、輝度計算には陰影計算モデルが必要である。代表的なものに Phong のモデル[BUIT75]、Torrance-Sparrowモデル[BLIN77]、Cookのモデル[COOK81]等が提案されている。これらの陰影計算モデルは、前述した隠面消去法と組み合わせて用いられる。しかし、これらの局所的照明モデル(local illumination model)では、光源の位置と物体表面の向きのみを考慮し、周りの環境を無視しているため、影、反射、透過、屈折等の現象を表現することができない。このため、隠面消去法に加え、これらの現象の様々な表現手法が提案されている[APPE68][WILL78][ATHE78][NEWE72][KAY79][BLIN76]。

視線探索法[WHIT80][OKAM81]は、光源から視点に至る光路を逆向きにたどって各画素の輝度を求める方法であり、周りの環境を考慮する大域的照明モデル(global illumination model)である。すなわち、隠面消去、影・反射・透過・屈折処理を一貫して処理することができる。各画素の輝度は、視点からその画素を通る半直線(視線)の方向からくる光の強度とし、視点に最も近い所で視線と交差する物体表面を求め(交差判定)、その点における輝度を計算する(輝度計算)。物体表面の輝度は、散乱光、鏡面反射光、透過光(このとき屈折する)の3成分からなるとする。影処理は、物体表面を仮想的な視点とし、そこから視線を各光源方向に向け、交差判定を行い、その間に存在する物体の透過率を光源強度に乗ずることで行う。反射・透過処理は、正反射・屈折方向にそれぞれ視線を向け、交差判定および輝度計算を行い、それぞれの方向からくる光の強度を求める(図2-1)。これらの処理を再帰的に呼び出すことで1画素の処理を行う。本来の視点(カメラ位置)から各画素に向けられた視線を1次視線、光源、あるいは反射・透過方向に向けられた視線を2次視線と呼ぶことにする。物体モデルとしては、視線との交差判定が可能で、交点における法線ベクトルが得られるものなら、どのような形式であってもよい。一般に、解が容易に求まる三角板や2次曲面等が多用されている[YOSH82]。

視線探索法の計算量は処理する視線の数に比例する。反射・透過のレベルを h 、光源の個数を L 、画素数を P とすると、全視線の数 R は、

$$R = P(2^h - 1)(1 + L)$$



$$I = \sum_{i=1}^N [(dic + drc \cos \theta_i) F + \cos^n \theta_{ri} \text{src}(\theta_r) J] L_i + \text{src}(\theta_r) l_r + \text{trc}(\theta_r) F l_t$$

- dic diffuse reflection constant for ambient light
- drc diffuse reflection coefficient
- src(θ) specular reflection coefficient
- trc(θ) transmission coefficient
- n glossiness
- I intensity of light leaving object's surface
- l_r intensity of light from direction of true reflection
- l_t intensity of light from direction of refraction
- L_i intensity of each light source
- F color of object
- J unit matrix

図 2-1 視線探索法と陰影計算モデル

となる。すなわち、画素数、光源数に比例し、反射・透過処理の多重度に対しては指数関数的に増大する。画面を構成する画素数は、通常 10^5 から 10^6 画素に及び、画素間のコヒーレンスが使えないため、膨大な計算量を必要とする。

また、この計算過程は、高々3次か4次の行列・ベクトル演算、もしくはスカラ演算だけから成り、処理の流れがデータに強く依存する。このため、ベクトル化率が低く、C R A Y - 1 [RUSS78]に代表されるパイプラインコンピュータや、G - P S Y C O [KUB080,83,84]に代表されるS I M D型 [FLYN72]マシン向き処理ではない。

しかし、各画素ごとに独立に計算が可能であることから、各画素の処理を並列に行うM I M D型 [FLYN72]のシステムが視線探索法による画像生成を高速化するには有効であると考えられる。しかも、各画素を処理するプロセッサ間で同期をとる必要がないため、システムを疎結合構成にすることができ、プロセッサ数に見合った安定した性能を得ることができる。

L I N K S - 1は、このような考察に基づき、高品位な画像を得られる一貫性のある画像生成アルゴリズムである視線探索法を高速に処理できるよう設計された。

2. 3 形状定義手法

スキャンライン法やZバッファ法等では、物体モデルとしては基本的に多角形のみ使用可能であり、複雑な形状は、微少な多角形の張り合わせとして表現される。このときスムーズシェーディング [BUIT75]を用いることにより、近似的に滑らかな曲面表現が可能である。曲面のモデリングを容易に行うためには、Bezier曲面、スプライン曲面等の手法 [ICHI79] [NEWM79] [YAMG82]を用い、レンダリングの際には微小多角形に分割してスムーズシェーディングを行うことが多い。

視線探索法の場合では、三角板、2次曲面を基本形状として用いる。三角板と視線との交差判定は、3次の逆行列を求めるだけですみ、2次曲面の場合は2次方程式を解くことに帰着される [YOSH82]。自由曲面は、スキャンラ

イン法等と同様、スプライン関数等で表現したもの[KURI87][FUJI88][KAWA88a]を三角板に分割、近似している(分割せずに、直接交差判定する方法もある[KAJI82])。

LINKS-1では、この他に、分布関数による曲面のモデリング手法を用いている[HIRA83][YAMN84][NISH85a]。これは空間中に有界な台をもつ関数を定義し、その関数値の線形和がしきい値と等しい境界を物体の表面とする手法である。この手法では、簡単なパラメータの変更だけでトポロジを考慮することなく、形状の連続的な変形が可能である。この変形過程は独特の視覚効果があり、数多くの作品に用いられている[KAWG83,84,85]。

映像化したいシーン全体は、これらの基本形状を階層的に組み合わせて構成する[KAWA82,84]。このとき、各階層での物体の存在領域を外接直方体で表わし、交差判定の高速化に役立てる[DEGU84b]。

以下に、LINKS-1において映像化に必要なデータをまとめる。

(1) 形状データ (shape data)

複数個の基本形状で物体またはその部品の形状を表現する。

(2) 構造データ (cluster data)

物体および部品の接続構造を表現する。物体は座標変換系を用いて階層的に構築される。シーン全体が世界座標系で、各階層(クラスタ)は局所座標系で定義される。その階層以下の物体が占める領域の外接直方体の情報をもつ。

(3) 属性データ (attribute data)

物体の光学的性質を表わすパラメータ(色、反射係数、透過率等)および貼付けデータとの接続情報からなる。

(4) 貼付けデータ (mapping data)

物体表面の模様(texture)や凹凸等を表現するための2次元配列状のデータ。

(5) 光源データ (lighting data)

光源の種類、強度、色、位置、向き等を表わす。光源の種類には、平行光線、点光源、スポットライト等がある。

(6) カメラデータ (camera data)

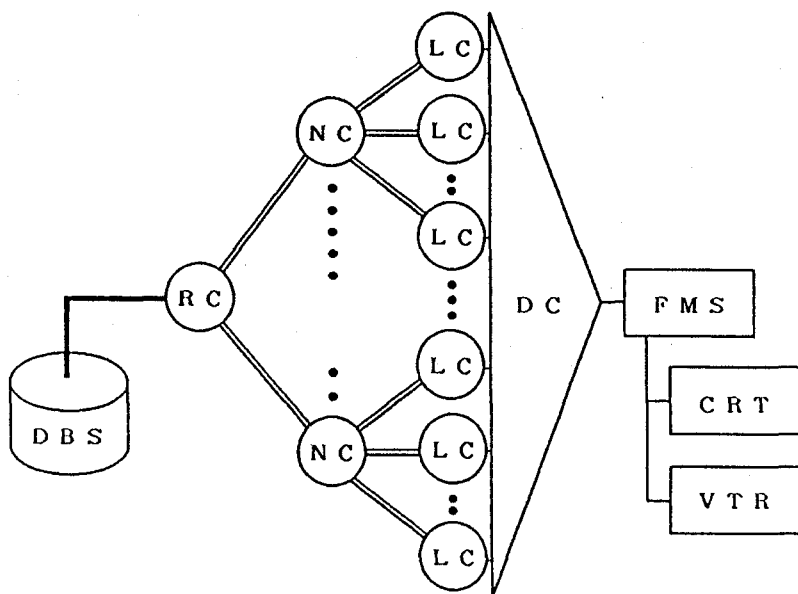
カメラの位置, 向き, ズーム角等を表わす.

これらのデータは, 言語的に記述するか[KAWA86b][FUKU86], モデリングシステムを用いて作成する[KAWA82,84]. 作成されたデータはデータベースシステムによって管理される[SHIM85][KAWA85b]. 動画を作成する場合は, これらデータの時間的変化を記述する必要がある[KAWA84].

2. 4 LINKS-1 のシステム構成

LINKS-1 は, 星状構造, ないしは木構造のMIMD型並列処理システムである (図2-2) [NAKA82][NISH83c][DEGU86]. 画像生成を行うLC (Leaf Computer), LCへの負荷の割り当てを行うRC (Root Computer) 及びNC (Node Computer) より構成される. RC, NC及びLCは同一構成のUC (Unit Computer) よりなる. UC間には, IMSU (Intercomputer Memory Swapping Unit) により結合されている. IMSUは, 別々のUCの主記憶の一部となる2組のメモリブロック (各々128KB) をバススイッチにより交換するもので, 非同期通信を実現する. LCにより計算された画素データは, DC (Data Collector) により集められ, FMS (Frame Memory System) に出力される.

UCは, Z8001 (クロック4MHz) をCPUとする制御ユニット (CU), 8086/87 (クロック5MHz) をCPUとする算術演算ユニット (APU), 1MBのメモリユニット (MU) の3つのユニットより構成されている. APUは, 主に算術演算を, CUはそれ以外の処理を行う. APUに単純な演算を行わせるとデータ転送のオーバーヘッドによりかえって速度が低下してしまうので, 各形状要素との交差判定, 輝度計算等の少ないデータ量の割に多くの演算を必要とする処理を行わせている.



DBS : Data Base System RC : Root Computer
 DC : Data Collector NC : Node Computer
 FMS : Frame Memory System LC : Leaf Computer
 = : Intercomputer Memory Swapping Unit

図 2-2 LINKS-1 のシステム構成

2. 5 LINKS-1 における並列処理方式

視線探索法においては、視線の 2 分木、すなわち物体との交点において反射方向と透過方向に分かれる有向 2 分木（視線探索木）の同一パス上にない視線に関しては全く独立に計算可能である。従って、視線探索法の並列化手法には、画面分割法と空間分割法[CLEA83][DIPP84]の 2 通りが考えられる。LINKS-1 では前者の画面分割法を用いている。空間分割法については第 6 章で述べる。

画面分割法では、画面を小画面に分割し、樹状に接続されたプロセッサの

葉 (leaf) の各プロセッサ (LC) に割り当てて処理を行う。

この手法では、各プロセッサの負荷ができるだけ均一になるように小画面を割り当てる必要がある。しかし、各小画面の処理時間を処理前に見積ることは、動画生成のようにフレーム間コヒーレンスを利用できる場合を除いて、かなり困難である。そこでLINKS-1では、次のような手法を用いている。画面をLCの台数より十分に多くの小画面 (合同) に分割し、各LCの処理要求に応じて小画面を分配した場合 (単純動的負荷分散法)、LCの処理時間のばらつき (最大-最小) は、各LCに割り当てられた最後の小画面の生成時間で抑えられる。すなわち、このばらつきは小画面の処理時間より小さくなるので、画面分割数を増し小画面を小さくすればよい。しかし、その反面、エリアコヒーレンスの利用効果の低下、処理要求回数 (分割小画面数) の増加に伴う通信のオーバーヘッドの増大のため、処理時間が逆に長くなる。従って、実際の画像生成では、ばらつきを決定する小画面が最後のものであることに着目して小画面の面積を一定にせず、処理の進行に伴い小画面を小さくする動的減少負荷分散法を用いている[NISH83a][DEGU84b]。

2. 6 LINKS-1における問題点

LINKS-1の今日までの運用経験や実験から、以下に述べるような問題点が指摘されている。

(1) 浮動小数点演算速度

C言語で書かれている画像生成プログラムをコンパイルし、その出力であるアセンブリ命令を解析した結果、単精度浮動小数点演算のアセンブリ命令数は画像生成プログラムの全アセンブリ命令のうち僅か8.7%にすぎないが、その処理時間は全処理時間の60%にも達する (表2-1) [KAWA85a]。この原因の一つは、制御用プロセッサZ8001と算術演算用プロセッサ8086/87両者間の通信のオーバーヘッドである。同様な問題が類似のシステムMC-1でも見られる[HIRA86]。

表2-1 画像生成プログラムにおけるアセンブリ
命令の分類と実行時間の比率

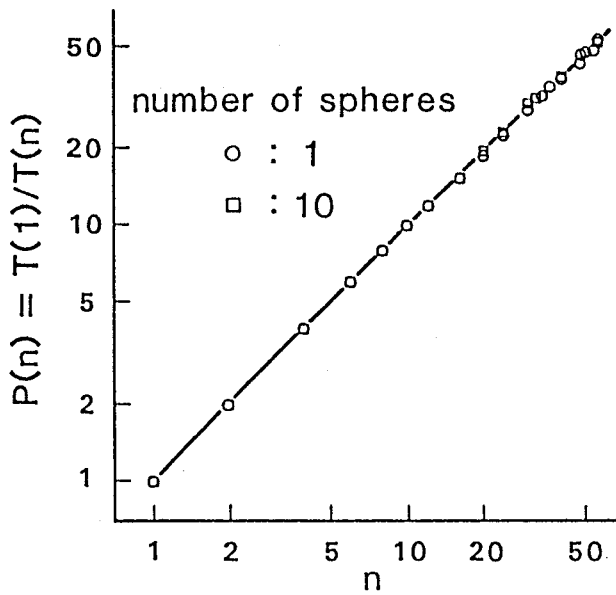
種類	数	比率(%)	時間(%)
データ転送	17963	56.8	20
浮動小数点演算	2729	8.7	60
アドレス計算	3261	10.3	10
その他	7645	24.2	10
合計	31598	100.0	100

(2) UC間通信方式

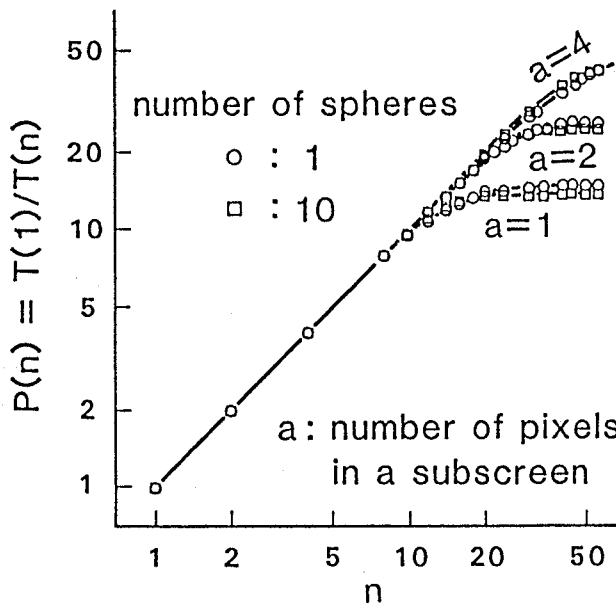
画面分割法では、各LCは基本的にすべての物体データを必要とする。LINKS-1では、LCのメモリにすべてのデータが書き込める場合は、処理開始後、初期転送の際に全物体データを各LCに転送する（バッチ転送方式）。この場合は、小画面の要求と割当て以外のUC間通信が全く無く、通信のオーバーヘッドが極めて少ないため、LCの台数に見合う処理速度の向上が達成された（図2-3(a)）[NISH85b]。ここでは、LCが n 台の時の画像生成時間を $T(n)$ とした。 $P(n)$ は、並列処理による画像生成の速度向上比（並列処理効率）を表わす。

一方、物体データ量がLCのメモリ容量を超える場合は、LCへの初期転送の際、形状データ、貼付けデータの転送を行わない。小画面の処理に必要な時点で、LCは上位のUCに要求し、これらのデータを転送してもらう（デマンド転送方式）。この方式では、データ量の制限は緩和されるが、通信のオーバーヘッドが増大するため、LCをある台数以上増やしても、それ以上処理速度が向上しなくなる（図2-3(b)）。この台数は、LCの内部処理時間とUC間の通信処理時間との比により決まる[NISH85b]。

LINKS-1において、画像生成のより一層の高速化を図るためには、画像生成プログラムの改良ないしは演算速度の向上、またはLC台数を増や



(a) バッチ転送方式の場合



(b) デマンド転送方式の場合

図2-3 LINKS-1の並列処理効率

す方法等が考えられる。しかし(1)より、演算能力の改善にはハードウェアの変更が必要であり、ソフトウェアの改良による高速化にも限界がある。LCの台数を増やすことは、(2)で述べたような問題があり、現実的には、LC64台程度の星状構造システム、もしくはLC256台程度の木構造システムが限界である。さらに多くのLCを有効に稼働させるためには、演算処理速度の向上より以上の通信処理速度の向上が不可欠であるが、この点についてもハードウェアの変更を必要とする。

2. 7 結言

本章では、高品位な画像を生成できる手法として視線探索法を取り上げ、これを用いた並列画像生成システムLINKS-1の概要と問題点について述べた。LINKS-1は、星状または木構造のMIMD型マシンであり、並列化手法として画面分割法を用いている。

映像制作におけるリアリズムの追求に伴い、画像生成処理時間、使用される物体データ量とも増加する一方である。シーンによっては、バッチ転送方式が使えないことも多い。デマンド転送方式では、LC台数を増やすことによる速度向上には限界がある。また、汎用プロセッサが用いられているLINKS-1のUCでは、画像生成処理向きに最適化された演算処理が行われているとは言い難い。従って、LINKS-1では今後、利用者の要求する、映像制作に必要な性能を提供できないと考えられる。元来LINKS-1は、本格的なCG専用システム開発のためのシミュレーション用システムである。以上の結果から、

(1) 単精度浮動小数点演算の高速化、

(2) UC間の通信機能の改良、高速化、

により、大規模な並列処理システムを構築し、実用化を図る。新しいUCは、高速化・安定化・小型化を図るためにゲートアレイカスタムVLSIを用いて実現する。

第3章 LINKS-2の ユニットコンピュータ

3.1 緒言

本章では、LINKS-1での実験結果、並びに前章での考察に基づき、新たに設計した並列画像生成システムLINKS-2のUCのハードウェア構成を中心に述べる。設計にあたり重視した点は、前述したように、演算処理速度およびUC間通信速度の向上である。LINKS-2のUCは、浮動小数点演算処理ユニット、アドレス演算処理ユニット、制御ユニット、UC間通信処理ユニット等から構成されており[KAWA85a,86a][OMUR86]、これらユニットは非同期に並列パイプライン動作する[KAWA88b]。演算の高速化を図るためには、UC内部のデータ転送をできる限り高速に行うことが重要である。このため、データメモリとプログラムメモリの分離、2項演算を1命令で実行するマイクロプログラム方式、データメモリのインタリーブ構成等の手法を用いた。乗算器については、論理機能素子の遅延時間に基づき計算機による自動設計を行った[KAWA85a]。UCの動作については第4章で、通信処理に関しては第5章で述べる。

3.2 ユニットコンピュータの設計方針

LINKS-1で稼働中の画像生成プログラムは、大量のデータに対し項数の少ない演算を繰返しており、配列や構造体の要素等複雑なアドレス計算を要するメモリ参照が極めて多い。従って、レジスタ割当ての最適化による演算速度の改善には限界がある上、メモリーレジスタ間のデータ転送が演算高速化の隘路となる。それ故、RISC[PATS81]に代表されるレジスタレジスタ演算主体のプロセッサ向きではない。

さらに、前述したように、スカラ、3次元ベクトル演算が主体で、処理の流れがデータに強く依存するためベクトル化率を上げることが難しく、ベクトルプロセッサ向きではない。

従って、簡素なハードウェアによる実現という観点から、スカラプロセッ

サを用い、特に多用されている整数の算術演算、論理演算、単精度浮動小数点の四則演算、開平、平方根の逆数演算、3次元ベクトル・行列演算、型変換を高速化することが望ましい。このためには、ベクトル化が困難な多数のデータをいかに演算器に供給し、結果を格納するかというデータ転送方式が、演算機構の方式以上に重要となる。

LINKS-2のUCでは、単一の演算器に高速に多量のデータを供給するため、アドレス生成、データ転送を極力並列化する。汎用レジスタをなくす代り、転送のオーバーヘッドを減らすことにより、全メモリをレジスタと同等に扱うことを可能にする。

以下の手法を用いて実現した。

(1) データ型は浮動小数点数、整数、ポインタ(すべて32ビット)のみとし、アドレス演算を単純化する。

(2) 2入力と1出力アドレスを含むマイクロプログラム方式[WILK69]とし、1命令で2項演算を実行する。プログラムは全てこの命令形式に展開される。

(3) 3組の32ビットバスを設け、メモリ、演算器にアドレス、データを並列供給する。

(4) アドレス演算専用処理ユニットを3組設けて、入出力アドレスを独立に計算する。

(5) データメモリとプログラムメモリを分離し、命令・データのフェッチを独立に行う。

(6) データメモリは4つのモジュールを用いたインタリーブ構成とし、クロスバースイッチを介して3組のバスと接続する。アドレス空間は32ビットリニアアドレスとした。

3.3 ユニットコンピュータの命令

3.3.1 データ型

LINKS-2のUCで扱うことのできるデータの型は、前述したように、浮動小数点数、整数、ポインタ(すべて32ビット長)であるが、これは

(1) アドレス演算が単純化できること、

- (2) アドレス空間が32ビットであること,
 - (3) 3次元画像生成処理では, 浮動小数点演算は単精度でも実用上差し支えないこと,
 - (4) 3次元画像生成処理では, 文字列処理は概して重要でないこと,
 - (5) システムの主要言語がC言語[KERN78]であること,
- 等の理由に基づく.

浮動小数点数は, 仮数部24ビット・指数部8ビットのIEEEフォーマット準拠, 整数は, 32ビット2の補数表現(T-int), 32ビット符号なし整数(U-int), 符号付31ビット絶対値表現(S-int)の3種類である.

3. 3. 2 命令形式

UCの制御は, 水平型マイクロプログラミング方式で行う. データメモリとプログラムメモリとを分離したので, 命令のビット幅は自由にとれる. 1命令は, すべて同一の形式で116ビット幅とした. 図3-1にマイクロプログラムの命令フォーマットを示す. 2項演算($\text{Src1 op Src2} \rightarrow \text{Dst}$)を1命令で行うため, 3つのオペランドフィールドが用意されている. また, それぞれのオペランドに対して独立にアドレッシングモードを指定できる. このような方式により, 32ビット整数及び浮動小数点数の加減乗除, 比較演算, 開平演算の他, 3次元および4次元(同次座標系)のベクトル演算(加減, 内積)等も1命令で行うことができる. 表3-1に主な命令の一覧を示す.

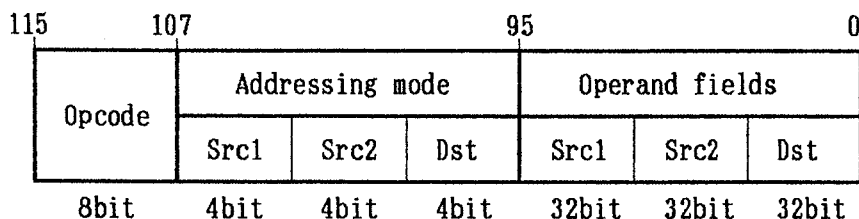


図3-1 LINKS-2の命令フォーマット

3. 3. 3 アドレッシングモード

命令で使用されるデータの実効アドレスは、3つのアドレッシングモードフィールドおよびオペランドフィールドから決定される。以下に述べるアドレッシングモードが用意されている。

(1) 直接アドレッシング (DA: Direct Addressing)

オペランドフィールドの内容が実効アドレスとなる。

(2) 間接アドレッシング (IA: Indirect Addressing)

オペランドフィールドの内容をアドレスとするメモリの内容が実効アドレスとなる。

表3-1 LINKS-2の主な命令

命令の種類	T-int	U-int	S-int	Float
加 算	ADD ADC ADDT		ADDS	ADDF
減 算	SUB SBC SUBT		SUBS	SUBF
乗 算	MULT	MULU	MULS	MULF
除 算				DIVF INVF
平方根				SQRTF ISQRTF
絶対値	ABST		ABSS	ABSF
ベクトル演算				VVM VLEN SVMVA SVMVS
比較演算	CMPT CJPT MINT MAXT	CMPU CJPU MINU MAXU	CMPS CJPS MINS MAXS	CMPF CJPF MINF MAXF
論理演算		AND OR XOR NOT		
シフト演算	LSAT RSAT	LSL LRC RSL RRC	LSAS RSAS	
型変換命令	CASTTS CASTTF	CASTUF	CASTST	CASTFT CASTFU
転送命令	MV			MVF
分岐命令	JP JPI JPR	CALL CALLI CALLR	RET RETI	

(3) イミディエイトアドレッシング (IM: Immediate Addressing)

オペランドフィールドの内容が直接、命令によって処理されるデータとなる。Src1, Src2に対してのみ有効である。

(4) インデックスアドレッシング (IX: Index Addressing)

インデックスレジスタ IXR_n ($n=0,1,2,3$) とオペランドフィールドの内容から実効アドレスが計算される。オペランドフィールドは、インデックスレジスタの指定が2ビットと16ビットのオフセットからなる。

① ノーマル (Normal)

IXR_n の値にオペランドフィールドのオフセットを加えたものが実効アドレスとなる。 IXR_n は変化しない。

② プリインクリメント (Pre-increment)

IXR_n の値にオペランドフィールドのオフセットを加えたものが、 IXR_n にセットされ、実効アドレスとなる。

③ ポストインクリメント (Post-increment)

現在の IXR_n の値が実効アドレスとなる。その後、 IXR_n の値にオペランドフィールドのオフセットを加えたものが、 IXR_n にセットされる。

④ ノーオペレーション (No operation)

IXR_n の値にオペランドフィールドのオフセットを加えたものが、 IXR_n にセットされるが、実効アドレスは計算されない。すなわち、メモリアクセスは発生しない。

同一の IXR_n が2つ以上のオペランドに使用されている場合は、Src1, Src2, Dstの順に、副作用が計算された後に次のオペランドの計算が行われる。

(5) 間接実効アドレッシング

(Indirect Indirect Addressing, Index Indirect Addressing)

(2), (4) ①, ②, ③の実効アドレスに対して、間接アドレッシングを行う。

(6) 無効 (Invalid)

それぞれのオペランドが存在しないことを示す。

表3-2にアドレッシングモードフィールド、図3-2にオペランドフィールドのフォーマットを示す。

表3-2 アドレッシングモードフィールドのフォーマット

I	X	M	アドレッシングモード	アセンブラ記法
0	0	00	Direct	address
0	0	10	Immediate	#data
0	0	11	Invalid	{ }
0	1	00	Index normal	$IXR_n \pm \text{offset}$
0	1	01	Index pre	$IXR_n \pm \text{offset}$
0	1	10	Index post	$IXR_n(\pm \text{offset})$
0	1	11	Index nop	$\{IXR_n \pm \text{offset}\}$
1	0	00	Indirect	(address)
1	1	00	Index indirect	$(IXR_n \pm \text{offset})$
1	1	01	Index indirect pre	$(IXR_n \pm \text{offset})$
1	1	10	Index indirect post	$(IXR_n(\pm \text{offset}))$

(ただし $n=0,1,2,3$)

DA, IA, IM mode

address/data

32bit

IX mode

0000000000000000	nn	offset
------------------	----	--------

14bit

2bit

16bit

offset: 2's complement(-32768~32767)

nn = 00: IXR0

01: IXR1

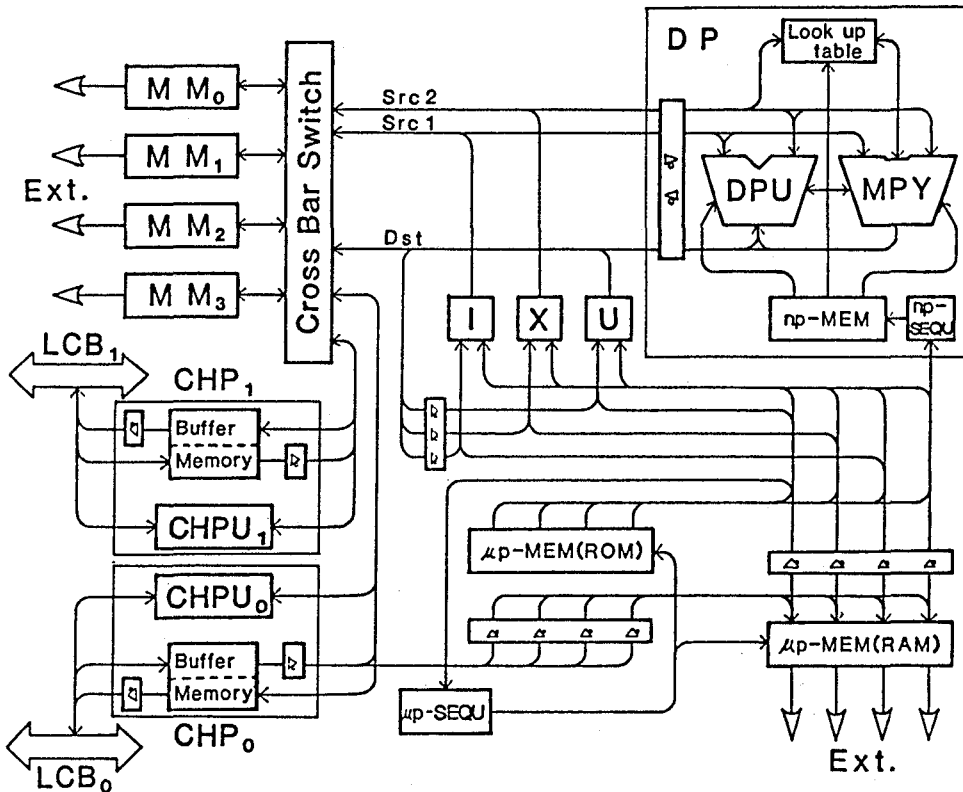
10: IXR2

11: IXR3

図3-2 オペランドフィールドのフォーマット

3. 4 ユニットコンピュータの構成

LINKS-2のUCのハードウェア構成を図3-3に示す。各ユニットの概要を以下に述べる。



- DP : Data Processor
- DPU : Data Processing Unit
- MPY : Multiplier
- IXU : Index Unit
- MM : Memory Module
- CHP : Channel Processor
- CHPU : Channel Processing Unit
- LCB : Links Communication Bus
- SEQU : Sequencer Unit
- μp-MEM: Micro Program Memory
- np-MEM: Nano Program Memory

図3-3 ユニットコンピュータのハードウェア構成

表3-3 ナノプログラムのフォーマット

np	制御線	制御線の機能	np	制御線	制御線の機能
87	NC	No Connection	43	MTY3	マスクタイプの設定
86	ROM2	ROMの選択	42	MTY2	
85	ROM1		41	MTY1	
84	ROM0		40	MTY0	
83	NC	No Connection	39	HALFE	ExALUへの命令
82	FS	MPYフェーズ選択	38	EXSINT	
81	MS1	MPYの 演算モード選択	37	EXCAIN	
80	MS0		36	EXMC	
79	NC	No Connection	35	EXS3	
78	OS2	MPYの 出力モード選択	34	EXS2	
77	OS1		33	EXS1	
76	OS0		32	EXS0	
75	IS3	MPYの B入力選択	31	SGT1	SgALUへの命令
74	IS2		30	SGT0	
73	IS1	MPYの A入力選択	29	SGADC	
72	IS0		28	SGMC	
71	NC	No Connection	27	SGS3	
70	RS2	MPYの 入力ラッチ選択	26	SGS2	
69	RS1		25	SGS1	
68	RS0		24	SGS0	
67	NC	No Connection	23	OVMASK	入力部に対する命令 符号操作 定数発生 SWAP制御 等
66	FLEN	フラグの制御	22	ABSA	
65	SELE	データの流れの制御	21	IMM1	
64	SELS		20	IMM0	
63	SEL6		19	ABSB	
62	SEL5		18	MINUS	
61	SEL3		17	NSWAP	
60	SEL2-1		16	MM_FA	
59	SEL2-0	ExALUへの 入力選択	15	TEA1	ExALUへの 入力選択
58	SEL1		14	TEA0	
57	R_EX	内部フリップ フロップの制御	13	TEB1	SgALUへの 入力選択
56	R_REI		12	TEB0	
55	ENE	入出力バッファの 制御	11	TSA1	SgALUへの 入力選択
54	ENS		10	TSA0	
53	ENA		9	TSB1	
52	ENB		8	TSB0	
51	STY1	ポストシフトへの 命令	7	SELA	演算データの選択
50	STY0		6	SELRA	
49	MTSEL1	出力部の制御	5	SELB	
48	MTSEL0		4	SELRB	
47	DEC3	命令コード	3	R_A1	内部レジスタの制御
46	DEC2		2	R_A0	
45	DEC1		1	R_B1	
44	DEC0		0	R_B0	

3. 4. 1 データプロセッサ

UCの演算処理は、データプロセッサ（DP：Data Processor）が行う。DPは、データプロセッシングユニット（DPU：Data Processing Unit）および乗算器（MPY）からなり、88ビット幅のナノプログラムと呼ぶより低レベルの水平型マイクロプログラムで制御される（表3-3）。図3-1のオペコードは、ナノプログラムの開始番地（10ビット）の上位8ビットを指す。

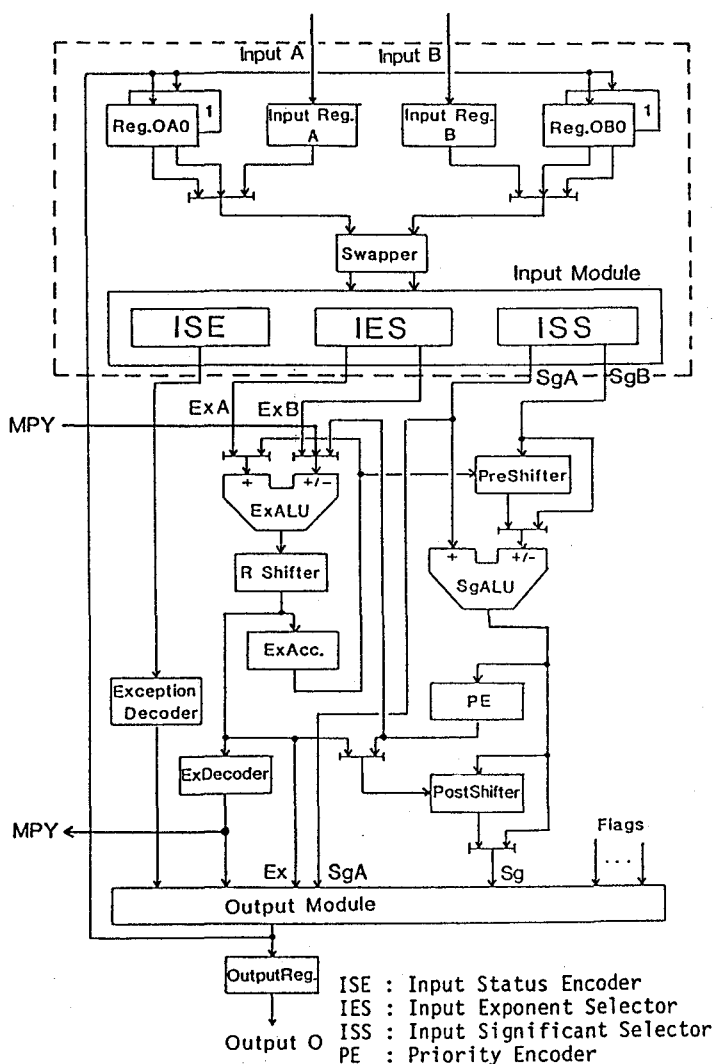


図3-4 DPUの機能ブロック図

DPUは、32ビット長の整数および浮動小数点仮数部の演算を行うSgALU、10ビット長の指数部演算を行うExALUを中心に構成されている。また、2項演算を行うための2本の入力と1本の出力（各々32ビット）、ならびにMPYとの通信線を持つ。この他、入力データより指数部、仮数部の選択を行う入力処理部、桁合わせのためのプリシフタ（Pre-Shifter）、正規化のためのポストシフタ（Post-Shifter）、零ディジットチェックを行うプライオリティエンコーダ（PE：Priority Encoder）、処理されたデータを集め例外処理等を施す出力処理部等の各モジュールが、図3-4のように接続されている。例外デコーダとExデコーダは、演算の結果、オーバーフローやアンダーフローが起こったとき出力を無限大や0等にマスクするためのものである。

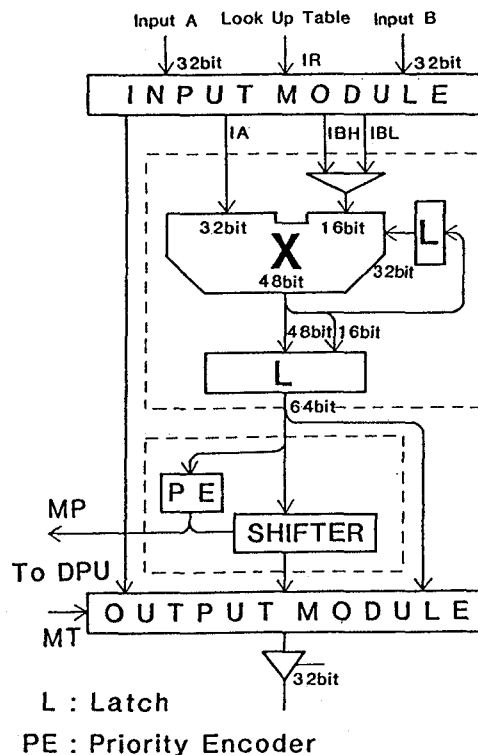


図3-5 MPYの機能ブロック図

MPYは、図3-5に示すように、通常の乗算のための32ビット幅の2本の入力ライン（IA, IB）、ルックアップテーブル参照用の入力ライン（IR）の他、例外処理時のマスク指定のためのDPUからの3ビットの入力（MT）を有する。32ビット×16ビット→48ビットの乗算回路と部分積ラッチを用いて、2フェイズで32ビット×32ビット→64ビットの乗算を行う。浮動小数点数の仮数部の乗算時に、正規化を行うため、プライオリティエンコーダ（PE）とシフタを持つが、指数部を計算しているDPUでの正規化のため、PEの出力はそのままDPUに渡される（MP）。また、乗算出力は32ビットである。

DPに用いられている高速化手法を以下にまとめる。

（1）SgALUとExALUを用いて、浮動小数点演算は仮数部と指数部の演算を並列に行う。

（2）加減乗算を高速化するため、逆数、開平、平方根の逆数演算は、DPのハードウェアに直結したルックアップテーブルを利用して求める。

（3）DP内にローカルバスおよびレジスタを持ち、ベクトル演算等DPUとMPY間でデータ転送を必要とする演算が通信のオーバーヘッドなく行える。

（4）MPYには、並列カウンタ型乗算回路[STNZ77]を用いた。設計の省力化と乗算の高速化のため、機能ブロックの遅延時間に基づき最適化した回路の生成を、計算機を用い自動で行った（3.5参照）[KAWA85a]。

表3-4 浮動小数点演算の処理時間の比較

命令	DP	i8087
加減算	0.33	21.0
乗算	0.3	23.6
除算	1.05	44.0
3次の内積	1.67	112.8
平方根の逆数	0.95	80.6

(μ sec)

表3-4に、DPの浮動小数点演算速度を、8086/87(クロック5MHz)と比較したものを示す(最小メモリアクセス時間を含む)。

3. 4. 2 インデックスユニット

インデックスユニット(I XU: Index Unit)の構成を図3-6に示す。I XUは、4つのインデックスレジスタ(I XR: Index Register)を内蔵し、命令のアドレッシングモード指定フィールドの制御によりアドレス計算を行う。比較的簡単なアドレッシングモードに対しては、3つの32ビット加算器を用いて Src1, Src2, Dstのアドレス計算を並列に行い、35nsでそれぞれのアドレスを同時に求めることができる。I XRに対する副作用によりアドレス計算に矛盾が生じないように、Src1, Src2, Dstの順に優先権が与えられている。より複雑なアドレス演算はDPを用いて行い、求められたアドレスは、Dstバスを通してI XUに入力される。

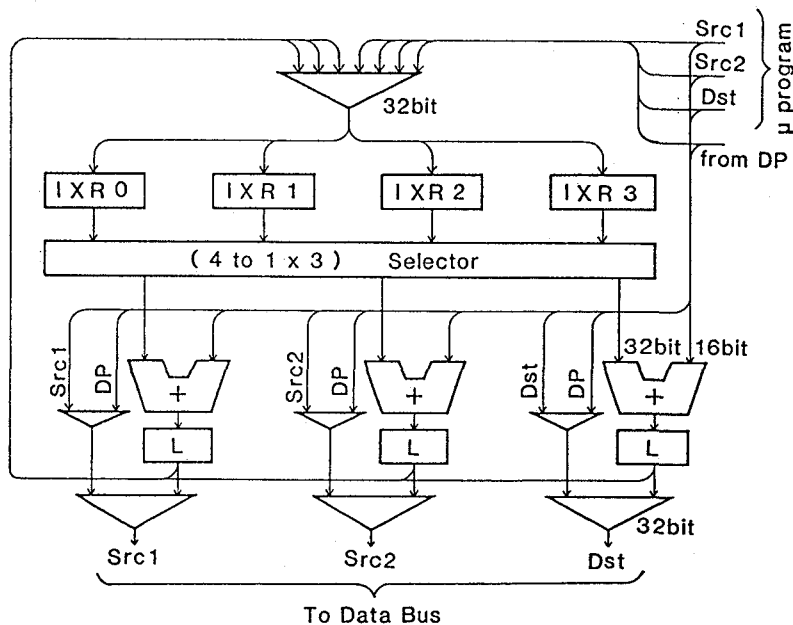


図3-6 I XUの機能ブロック図

3. 4. 3 クロスバースイッチとデータメモリ

各データバス(Src1, Src2, Dst)はクロスバースイッチ(CBS: Cross Bar Switch)を介してデータメモリと接続されている。データメモリは、インタリーブされた1MB(256KW, 拡張可能)のメモリモジュール(MM: Memory Module) 4組から構成されている。

データメモリは、CBSを介して2つのUC間通信専用プロセッサ(チャネルプロセッサ: 後述)とも接続されており、UC間的高速データ通信が可能である。CBSには、バイトスワップ機能があり、他のバイトスワップを行うプロセッサとの接続も容易である。

3. 4. 4 シーケンサユニットとプログラムメモリ

シーケンサユニット(μ p-SEQU: Micro Program SEQuencer Unit)は、マイクロプログラムメモリ(μ p-MEM: Micro Program MEMory)からの命令の取り出し、各種のハンドシェーク信号の制御等、UC全体のタイミングを管理している。

μ p-MEMの容量は、高度な演算命令が1命令で用意されており、また転送命令が少なくすむので、64KW(116ビット幅)とした。この容量は、画像生成プログラム全体をロードできる容量である。

3. 4. 5 チャネルプロセッサ

チャネルプロセッサ(CHP: CHannel Processor)は、DPとは独立に動作するUC間通信専用プロセッサであり、図3-3に示すように、チャネルプロセッシングユニット(CHPU: CHannel Processing Unit)と2KWのバッファメモリ(BM: Buffer Memory)から構成されている。外部バスの仕様、通信手順等は第5章で詳しく述べる。

3. 5 乗算回路の自動構成手法

並列カウンタ型乗算回路の加算回路は，全加算器（3 入力 2 出力カウンタ）を樹状に結線し部分積を 2 つに減らす．最後に桁上げ先見回路（CLA）を用いて積を求める．本手法では，この結線を桁上げの遅延時間が小さくなるように自動配線する．すなわち加算回路における各加算器の入力を各信号の確定時刻に基づき選択することにより乗算回路を構成する．

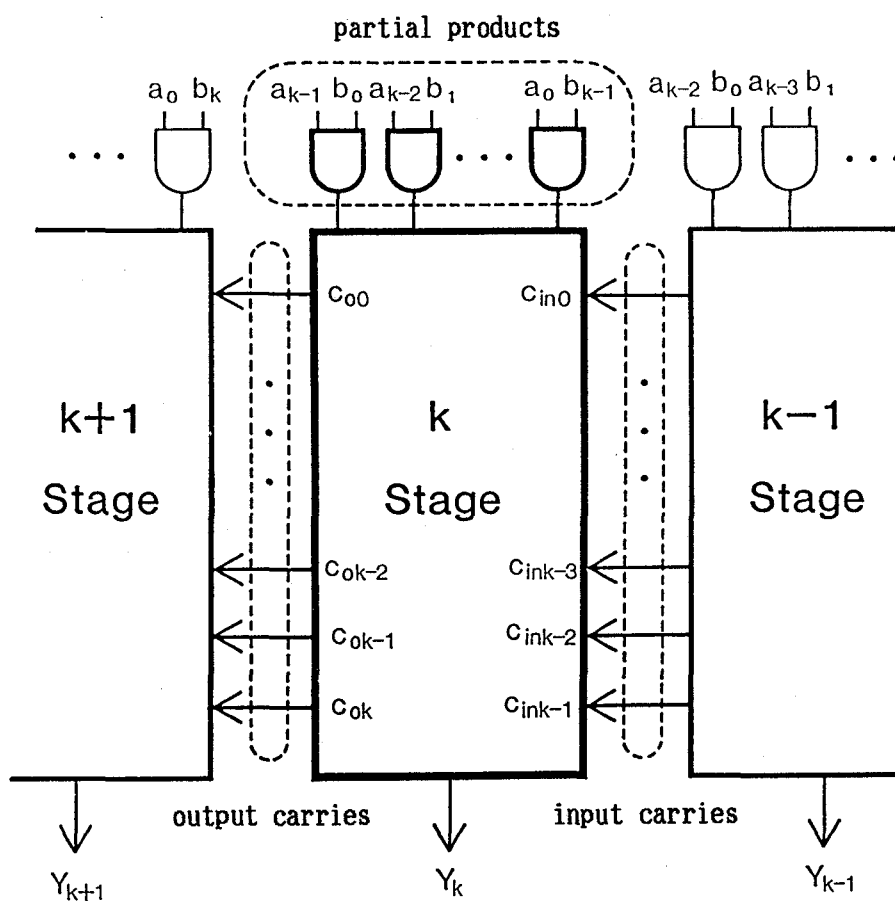


図3-7 部分積項の加算に対するステージのモデル

積のkビット目を求める作業は、図3-7のようにモデル化できる。すなわち、各部分積項と前段からの入力桁上げ群を加えて1ビットの加算出力と次段への出力桁上げ群を作成することである。これをステージと呼ぶことにする。ステージの入力に注目すると、部分積項は乗数と被乗数が与えられた次の瞬間（2△後、但し△は論理素子一段の遅延時間）に確定するため、桁上げの伝搬遅延が支配的になっている。従って、各ステージにおいて値の遅く確定する桁上げほど下位の全加算器に入力すればよい。乗算回路全体の結線の仕方を、計算機を用いた総当り法により決定することも考えられるが、全加算器の遅延時間はファンアウトや値の変化の方向等により変わること等から、多ビット長の乗算回路では計算時間が急速に増大し、現実的ではない。

そこで、本構成法では、下位のステージから順に乗算回路を最適構成していく。すなわち、第kステージを最適構成した後、その出力桁上げ群の遅延時間に基づいて、第k+1ステージを構成する。ここでいう最適構成とは、単にそのステージの加算出力が最短時間で得られる方法を指すのではなく、出力桁上げ群の遅延時間を最小にすることである。しかし、本構成法では、出力桁上げ群の遅延時間を直接評価せず、各加算器を最大限に有効利用する方法を用いて間接的に最適構成する。例えば、全入力から出力までの遅延時間が4△である加算器の入力が同時に確定したとすればこの加算器は有効に使われていると見なされる。もしこの入力のうち1本が他の2本に5ns遅れて確定したとすればこの加算器は5ns分無駄に使われていると見なされる。すなわち、入力信号線の確定時刻の関数として評価が行われる。ここではこの評価値をe値と呼ぶことにする。

e値は、乗算回路の実現に使用したCMOSゲートアレイの機能ブロックに含まれる全加算器をモデルとして、ここでは次のように決定した。

$$e = 3t_s - t_{ci} + 2.6t_{co}$$

ここで、 t_s 、 t_{ci} 、 t_{co} は、それぞれ加算器出力S、桁上げ入力C_{in}、桁上げ出力C_{out} が確定する時刻である。

あるステージに入力される入力桁上げ群と部分積項の集合をMとし、出力桁上げ群の集合をCとする。MおよびCの各要素はそれが確定するまでの遅

延時間が既知である。以下に構成アルゴリズムを述べる。

(1) そのステージの部分積項の確定時刻を計算してMの要素に加える。またCを空にする。

(2) Mの中から最も遅延時間の短い要素を一つ取り出しこれをaとする。

(3) Mの中から二つの要素を全ての組み合わせで取り出し、aとのe値が最も小さい(評価の最も高い)ものを選び、これを(b, c)とする。このとき(a, b, c)はMの中から既に除かれている。

(4) この(a, b, c)を一つの全加算器で処理する。得られた和出力の確定時刻を計算してMの中に戻す。また桁上げ出力は同様に確定時刻を計算してCに加える。

(5) Mの要素が一つになるまで(2)から(4)を繰り返す。このMに残った最後の一つがこのステージにおける積出力である。

(6) Cの要素をMの要素とし、次のステージの処理を(1)から行う。

3. 6 結言

本章では、LINKS-2のUCの構成について述べた。画像生成処理における演算の特徴を分析した結果、本UCでは、プログラムメモリとデータメモリの分離、データメモリのインタリーブ構成、2入力と1出力アドレスを含む水平型マイクロプログラム方式等を採用している。これらの方式により、2項演算を始めとする演算処理に伴う命令のフェッチ、アドレス計算、データの読み書き等を、各内部ユニット間で非同期に並列処理することができる。

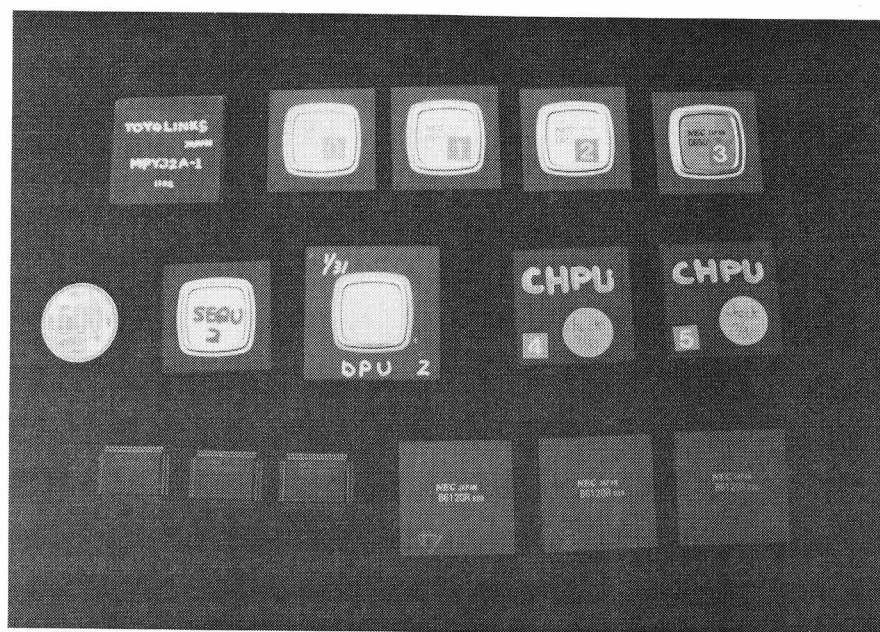
UCの各ユニットは、システムの小型化、高速化、安定化のため、ゲートアレイVLSIを用いて実現した。各ユニットに使用したゲートアレイの一覧を表3-5に、完成したチップを図3-8に示す。

本UC1台は、PUボードとMUボードからなる。PUボードには、DP, IXU, μ p-SEQU, μ p-MEM, CHP0 が、MUボードには、データメモリ, CBS, CHP1 が実装されている(図3-9)。また、UC4台が収容できる筐体を図3-10に示す。これらの基板は、現在、動作試験中

である。

表3-5 使用ゲートアレイ一覧 (UC 1台)

ユニット	種類
DPU	CMOS 10000ゲート×1
MPY	CMOS 8000ゲート×1
IXU	TTL(内部ECL)2000ゲート×3
CBS	CMOS 5000ゲート×4
μp-SEQU	CMOS 4000ゲート×1
CHPU	CMOS 10000ゲート×2



上段左からMPY, CBS×4, μp-SEQU, DPU, CHPU×2, 32ビットバスバッファ×3, IXU×3

図3-8 使用したゲートアレイVLSIチップ (UC 1台分)

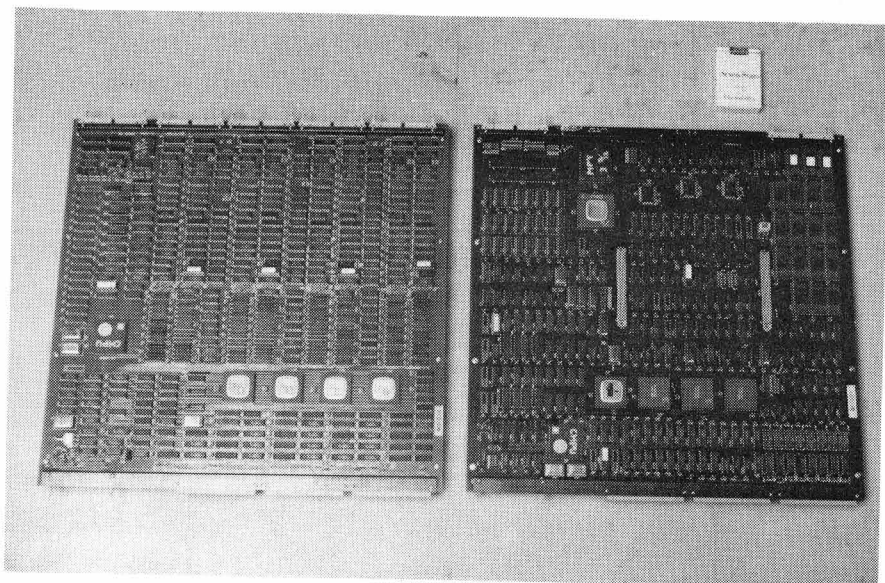


図3-9 PUボード(右)とMUボード

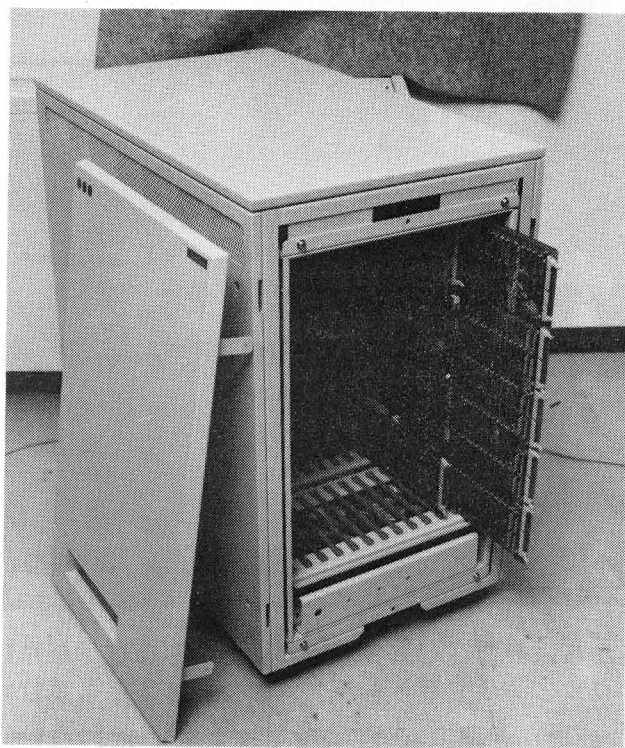


図3-10 LINKS-2の筐体(UC4台収容)

第4章 演算処理方式とその評価

4.1 緒言

第3章で述べたように、画像生成処理における演算を高速化するためには、UC内部でのデータ転送を高速化することが重要である。LINKS-2のUCでは、4ウェイにインタリーブされたデータメモリをクロスバースイッチ(CBS)を介して3組のバスと接続している。また、アドレス演算処理専用インデックスユニット(IXU)を3組用いている他、2入力1出力アドレスを1命令中に含むマイクロプログラムでUCの制御を行っている[KAWA85a,86a]。これらの方式により、演算に必要なアドレスやデータは、できる限り並列に転送、処理される。また、処理の流れは、パイプライン化されている。本章では、このデータ転送方式の概要とユニットコンピュータの動作について述べる。また、シミュレーションにより、その演算処理能力並びに高速化手法の効果を明らかにする[KAWA88b]。

4.2 データの配置

図4-1に示すように、3次元ベクトルの各成分を順に4つのモジュールに分配すれば、連続した4ベクトルの同一成分は全て異なるモジュールに置かれる。従って、局所的に定義された3次元ベクトル間で演算を行う場合、被演算成分の演算器への供給及び演算結果の格納が並列化できる確率が高くなる。但し演算結果については、処理をパイプライン化することで、その格納を次の被演算成分の供給と同時に行う。

4つのモジュールに等確率でランダムに配置されたスカラデータに対し2項演算を行った場合でも、平均1.69回のアクセスで演算を終了できる。加えて、残り1つのモジュールのアクセスが競合なく行えるので、演算処理と並行してUC間のデータ転送が効率よく行える。

アドレスは、4つのモジュールにインタリーブして割り付けた(図4-1)。アドレスの下位2ビットがモジュールを選択する。C言語の自動変数は、スタック上にとられ、実行時にそのアドレスが決定される。このため、一般に

変数の各モジュールへの実行前の配置決定は困難である。しかし本方式では、配置に関する特別な配慮を必要としない。すなわち、コンパイラはデータをアドレス順に配置するだけでよい。

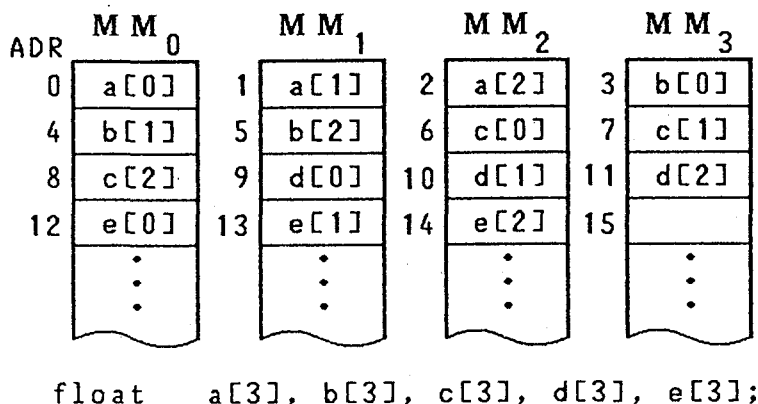
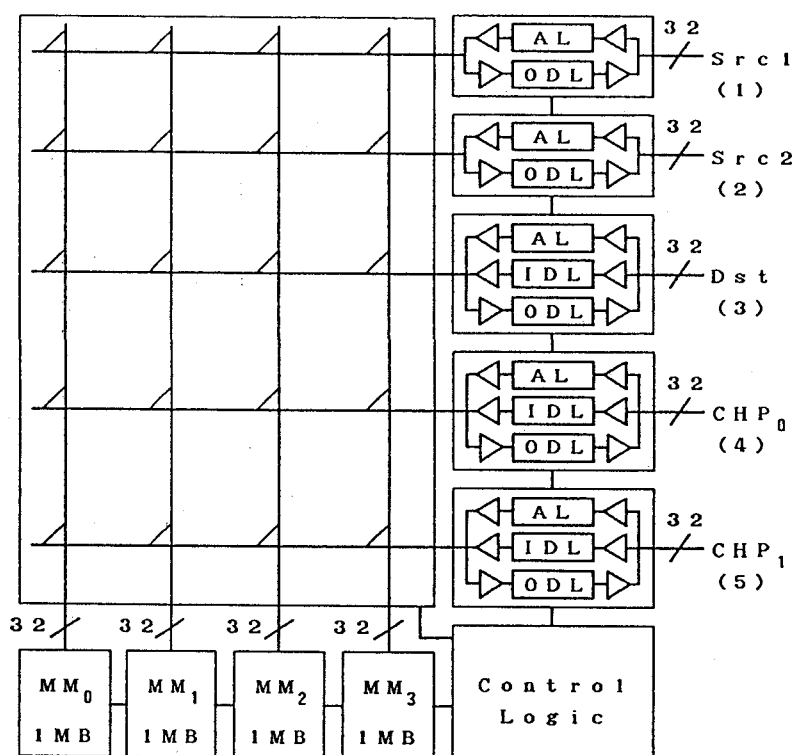


図 4-1 3次元ベクトルのデータメモリ上への配置

4. 3 クロスバースイッチの制御

各MMは、図 4-2 に示すCBSにより、DPとの3本のバス(Src1, Src2, Dst), 2つのCHPと接続されている。CBSはアドレス、データラッチを持ち、送信側ユニットはメモリアクセス完了を待たず次の処理に移れる。

5組のうち何れかからアクセス要求が到着次第、要求モジュールとの接続が行われるが、既に通信中であれば待たされる。アクセス要求が競合した場合は、Src1, Src2, Dst, CHP0, CHP1 の順にラウンドロビン方式で決定する。



AL : Address Latch IDL : Input Data Latch
 ODL : Output Data Latch () : Priority

図4-2 クロスバースイッチとデータメモリの構成

4. 4 UCの動作

UCの各ユニットは独自のクロックで非同期に動作し、ハンドシェークによりデータの授受を行い、必要なデータ・条件が揃えば自己の処理を他と独立に行う。これらのハンドシェークは主として μ p-SEQUの管理下で行われる。

単純なアドレッシングモードの2項演算の場合、まず μ p-MEMから、オペコードがDPに、オペランドアドレスがIXUに送られ、アドレス計算が行われる。次にそのアドレスがCBSを介してMMに送られ、読み出された

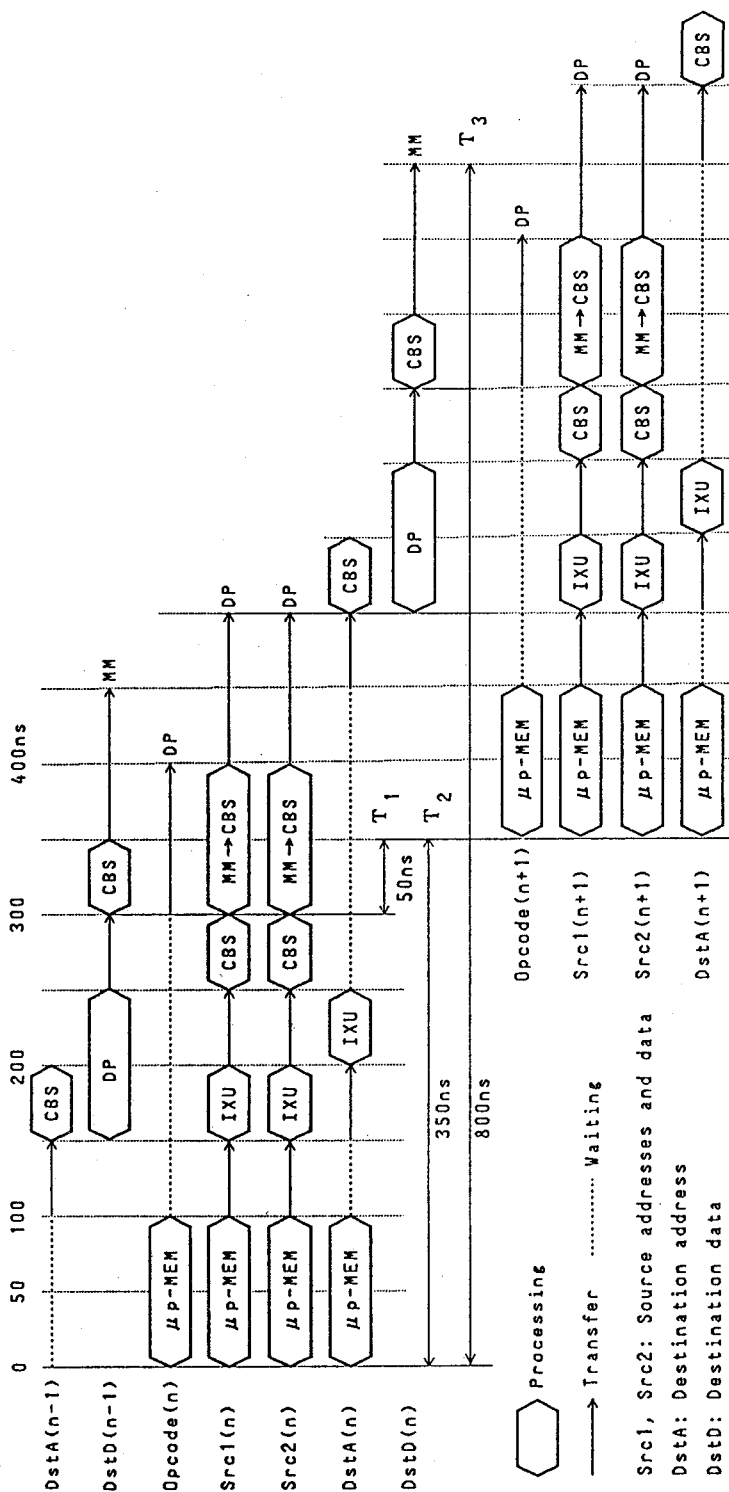


図 4-3 2 項演算の並列パイプライン処理 (メモリアクセス競合のない場合)

データはCBSを介してDPに送られる。演算結果は再びCBSを介しMMに戻される。これらのシーケンスは、並列化されているだけでなく、パイプライン化されている。図4-3に、MMのアクセス競合のない場合における $n-1$, n , $n+1$ 個目の命令実行の様子を、50nsを単位として、各ユニットでの処理時間、待ち時間、データ転送時間を含めて時間を追って示した。ここでは、オペコードのDPへの転送開始から50ns後に次の命令の準備を始めるとした(図4-3 T1)。

本UCでは演算に際し、ユニット間のデータ依存関係に矛盾が生じないよう以下に述べる命令実行上の制約を設けている。

- (1) μp -MEMは、1つの命令のオペコードと3つのアドレスを全て送信した後に初めて次の命令の準備を開始する。
- (2) CBS(Src1, Src2)は、アドレス受信後、DPへのデータ転送を完了するまでバスを確保し続ける。
- (3) CBS(Dst)は、必要なMMを獲得後、演算結果を書き込むまで確保し続ける。

4. 5 演算処理方式の評価

本UCの演算処理能力を調べるため、画像生成プログラムのうち輝度計算モジュールのオブジェクト(マイクロプログラム)を、UCの命令レベルのシミュレータ上で実行した。このシミュレータは、LINKS-2用プログラムの検証を目的として作成され、MMのアクセス競合の状況と実行された命令のサイクル数からその実行時間を求めることができる。アクセス競合の状況を表4-1に、実行時間を表4-2に示す。実行時間については他の機種での実行結果と比較した。実行命令の総数は262命令、よって1命令の平均実行時間326ns、スループットは3.1MIPSとなった。但し本UCの命令は、他のプロセッサの命令と粒度が異なり、単純にMIPS値のみを比較することはできない。また、競合のないメモリアクセスの回数は80%を超え、データメモリのインタリーブ構成を始めとするデータアクセスの並列化手法が画像生成処理に適したものであることがわかる。

表 4-1 輝度計算におけるMMのアクセス競合

競合の種類	比率(%)
競合なし	85.6
Src1-Dst	6.0
Src1-Src2	3.4
Src2-Dst	2.2
前のDst-Src2	1.6
前のDst-Src1	0.7
Src1-Src2-Dst	0.5

表 4-2 輝度計算の処理時間

機種	処理時間(msec)	速度比
VAX11/780 FPA付	0. 7 4 4	5. 6
LINKS-1 UC1台	4. 1 7	1. 0
LINKS-2 UC1台	0. 0 8 5 4	4 8. 8

次にパイプライン化の効果を、4. 4で述べた2項演算におけるUCの動作モデルを例に解析した。図4-3から、MMのアクセス競合のない場合、1命令の実行時間は800ns(図4-3 T3)、パイプラインの周期は定常状態では350ns(図4-3 T2)、従って並列度は2.29、スループットは2.9MIPSであることが導かれる。さらにこの図から、命令実行時間・パイプライン周期を決定しているクリティカルパスは、Src1, Src2の μ p-MEMからDPに至るパスであることがわかる。

MMのアクセス競合のパイプラインに対する影響を調べるため、このモデ

ルを計算機上に構築し、全く競合のない場合、表4-1に示した頻度で発生した場合についてシミュレーションを行った。その結果を表4-3に示す。競合のない場合は前述の解析結果とよく一致している。全く同じでないのは、パイプラインの立上りの過渡状態を含むためである。競合のない場合とある場合のパイプラインの平均周期の差は僅か5nsしかない。これは、

(1)競合頻度の高いSrc1,2とDstは実際には同時刻にアクセスが起こらない、
(2)CBSの利用率(MMのアクセスを含む)が84~85%と余裕があるため、と考えられる。ここで利用率とは、各ユニットの内部処理時間とデータ転送時間の和の平均周期に対する比であり、純粋な待ち時間を除いた稼働率を表わす。この実験は簡単な2項演算のモデルであるが、画像生成処理ではMMのアクセス競合が与える影響は少ないといえる。DPの利用率も70%を超え、効率のよいデータ供給が実現されている。

マルチプロセッサ構成の場合、CHPからのアクセス要求による競合が発生するが、その影響はシステム構成および実行制御方式に大きく依存する。ここでは問題を単純にするため、CHPからのアクセス競合の影響を無視できると仮定した。

表4-3 UCのパイプライン動作のシミュレーション結果

	競合なし	競合あり
1 命令の平均実行時間(ns)	800.7	810.4
パイプラインの平均周期(ns)	351.0	356.0
パイプラインの並列度	2.28	2.28
スループット(MIPS)	2.85	2.81
μp-MEM(opcode)の利用率(%)	57.0	56.2
μp-MEM(その他)の利用率(%)	42.7	42.1
I X Uの利用率(%)	42.7	42.1
CBS(含MMアクセス)の利用率(%)	85.5	84.3
DPの利用率(%)	71.2	70.2

4. 6 結言

本章では、演算データのデータメモリ上への配置手法、CBSの制御方式について述べ、簡単な2項演算の場合において、その処理方式を明らかにした。本UCでは、内部の各ユニットは、非同期に並列パイプライン動作し、ハンドシェイクによりデータの授受を行う。このモデルを計算機上に構築し、データメモリのアクセス競合が演算速度にどのような影響を与えるかシミュレーションを行った。その結果、輝度計算時に起こるアクセス競合はかなり少なく、演算処理に与える影響は小さいことがわかった。

第5章 ユニットコンピュータ間 通信方式とその評価

5.1 緒言

デマンド転送方式における並列処理効率の低下を防ぐためには、競合のない専用バス方式が優れているが、IMSUの場合にも見られるように、接続する台数に比例して接続線数、UC1台当りの通信用ハードウェア量が増大する。

一方、共有バス方式では、システム構成のソフトウェア的な制御が容易であり、耐故障性、負荷分散法の改良等の点で優れるが、バス競合は避けられない。しかし、データ転送が十分高速であれば、共有バス方式でも並列処理効率の低下を防ぐことができる。実際、LINKS-1の星状、木構造システムによる並列画像生成では、1対多の接続が基本であり、RCが1つのNC、LCと通信中は、他のNC、LCは待たされており、等価的に共有バス方式に置き換えることができる。

そこで、LINKS-2では、システム構成の柔軟性を重視し、32ビット共有バス(LCB: Links Communication Bus)を用いて高速ブロック転送を行う。本章では、LCBを用いたUC間通信方式の概要について述べ、並列処理システムの構成例を示す。また、星状構造システムの場合の並列処理効率をシミュレーションによって求め、LINKS-1の場合と比較を行う[KAWA88b]。

5.2 UC間通信方式の特徴

本方式は、次のような特徴を持つ。

(1) LCBにおけるバス競合等が演算処理に及ぼす影響をできる限り減らすため、DPとは全く独立して動作する通信処理専用プロセッサ(CHP: Channel Processor)を用いて制御する。

(2) データメモリへのアクセス競合を最少限にし、両者の並列稼働率を上げるため、CHPはクロスバースイッチを介してデータメモリと接続する。

(3) 多様なシステム構成を実現するため、UCは2つのCHPを持ち、2系統のLCBと接続できる。

(4) バス調停のための集中的な制御機構は用いず、各CHPが分散して行う方式とした。

(5) データ転送とは別にUC間の動作制御(割込、リセット等)を実現した。

(6) 外部装置との通信の高速化のため、外部装置側にも、同等の機能を有する通信制御プロセッサ(I/Oプロセッサ)を持たせ、LCBに直接接続できるようにした。

(7) 大規模な並列処理システム構築のため、バス拡張用バッファを用意した。1本のLCBに接続可能な台数は、バス利用率にのみ制限される。

(8) LCBの転送速度は、最高25MW/sec(100MB/sec)とした。

(9) LCBに接続されている任意のCHP間での1対1通信、全CHPに対するブロードキャスト等各種通信モードを持つ。

5.3 チャンネルプロセッサの構成と機能

CHPは、図5-1に示すように、バス調停機構(Bus Controller)、通信制御機構(Transfer Controller)と、それらを制御するシーケンサ(CHP Sequencer)を持ったチャンネルプロセッシングユニット(CHPU: Channel Processing Unit)と、2KWのバッファメモリ(BM: Buffer Memory)から構成されている。同一LCB上のCHPUには、識別用にそれぞれ固有のモジュールナンバー(MN: Module Number)が与えられている。

CHPUの機能は、

- (1) LCBの制御と調停、
 - (2) LCBとBM間のデータおよびプログラム転送、
 - (3) MMとBM間のデータ転送、
 - (4) μ p-MEMとBM間のプログラム転送、
- 等である。

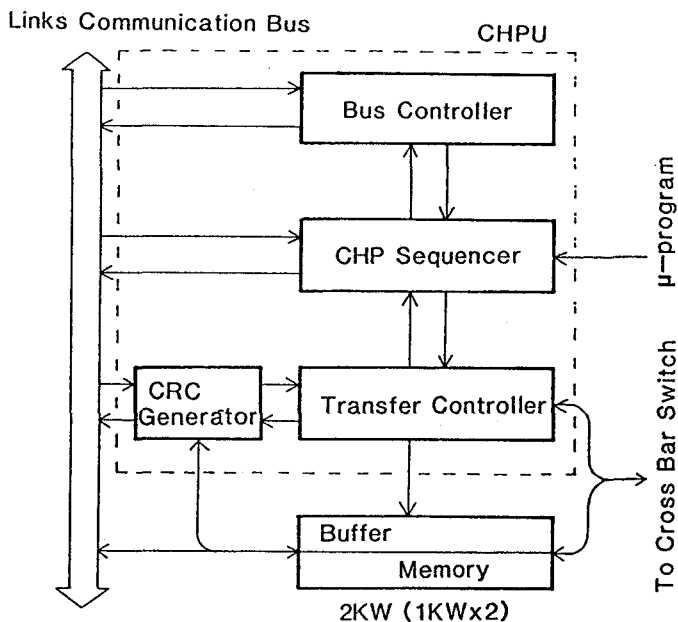


図 5-1 CHPの機能ブロック図

またCHPUは、MN等通信に必要な情報を格納するため、内部にメモリ空間にマッピングされたレジスタを持つ。その一つであるコマンドレジスタにマイクロプログラムや他のCHPUから命令を書き込むことにより、CHPUを起動する。起動後は、CHPUはDP等のUCの他のユニットと並行して動作する。CHPUの内部レジスタを表5-1に示す。

図3-3に示すように、データメモリはCBSを介してBMと接続されている。UC間のデータ転送はすべてこのBMによりバッファリングされ、演算処理とUC間通信におけるデータメモリのアクセス競合を減らしている。

表 5-1 CHPUの内部レジスタ

レジスタ名	保持する内容
WR 0	MN, イニシャルマスタCHPUの設定
WR 1	送信先のMN
WR 2	送信先へのメッセージ
WR 3	転送データのサイズ
WR 4	転送データブロックの先頭アドレス
WR 5	受信先におけるデータブロックの先頭アドレス
WR 6	コマンドレジスタ
WR 7	パラメータレジスタ
RR 0	LCBにおける最大のMN + 1
RR 1	LCBにおける最小のMN
RR 2	データ転送元のMN
RR 3	データ転送と共に送られてきたメッセージ
RR 4	内部ステータス

WR : 書き込み専用レジスタ

RR : 読み出し専用レジスタ

5. 4 LCBの状態遷移とUC間通信手順

一本のLCB上では、常に何れか一つのCHPUが、バスを占有している(マスタCHPU)。システム立ち上げ時のマスタCHPUを、イニシャルマスタCHPUと呼ぶ。イニシャルマスタCHPUは、各LCBにつき1台予め指定されている。データ転送時の送信側CHPUをトランスミッタ、受信側CHPUをレシーバと呼ぶことにする。

データを転送するためには、まずマスタCHPUとなりバスを獲得する必要がある。以下に、バス獲得並びに通信手順を、LCBの状態遷移の観点から示す。

(1) リセット状態 (Rs)

システムリセットによって発生し、すべてのCHPUのMNや通信速度の設定等、内部の初期化を行う。終了後、イニシャルポーリング状態に移行する。

(2) イニシャルポーリング状態 (Pi)

対象となるLCBに接続されたCHPUのうち最小、最大のMNを持つものを、二分探索法をハードウェア化したポーリングにより調べる。ポーリングはLCBごとに予め指定されたイニシャルマスタCHPUによって行われる。求められたMNは全てのCHPUにブロードキャストされ、通常ポーリング時に利用される。終了後、イニシャルマスタCHPUは、マスタ権を引続き保持しアイドル状態に移行する。

(3) アイドル状態 (Id)

マスタCHPUが各CHPUからのバス要求を待っている状態である。新たなバス要求を出すことができるのはこの状態の時だけである。バス要求が発生すると通常ポーリング状態に移行する。

(4) 通常ポーリング状態 (Pn)

マスタCHPUは、イニシャルポーリング状態と同じハードウェア化した二分探索法を用いて、バス要求を出しているCHPUの中から1つを特定し、そのCHPUにマスタ権を譲る。マスタCHPU自身がバス要求を出していた場合、ポーリングを行わず、引続きマスタ権を保持する。マスタ権を得たCHPUは、トランスミッタとして通信のための処理を開始し、通信準備状態に移行する。ポーリングは、最悪 $100 \times \lfloor \log_2 n \rfloor + 300(\text{ns})$ で終わる。この時、マスタ権を獲得できなかったCHPUは、引続きバス要求を出し続ける。

(5) 通信準備状態 (Tp)

トランスミッタが、レシーバを指定し、通信を始めるまでの状態である。この時、複数のレシーバを指定することができ、自分を除いたすべてのCHPUを指定することもできる（ブロードキャスト）。指定したレシーバが受信可能であれば、通信実行状態に移行する。指定したレシーバのうち少なくとも1つが受信不可能で、かつバス要求を出し続けているCHPUがあれば通常ポーリング状態に移り、なければアイドル状態に戻る。通常この状態は、300nsで終わる。

(6) 通信実行状態 (Te)

トランスミッタのMNと転送データのサイズからなるヘッダ、3ワードの

CRC誤り検出符号からなるフッタを付加したパケット方式で転送を行う。誤りが検出されると、トランスミッタは通信を中断するか、もしくは再転送を行う。この際、自動的に転送速度が落とされ、それ以降の通信においてもこの転送速度が採用される。誤りがなければ、レシーバはアクノリッジ信号を返す。これを受けたトランスミッタは、バスに終了信号を送出して通信を終了する。この時、バス要求を出し続けているCHPUがあれば通常ポーリング状態に移り、なければアイドル状態に移行する。

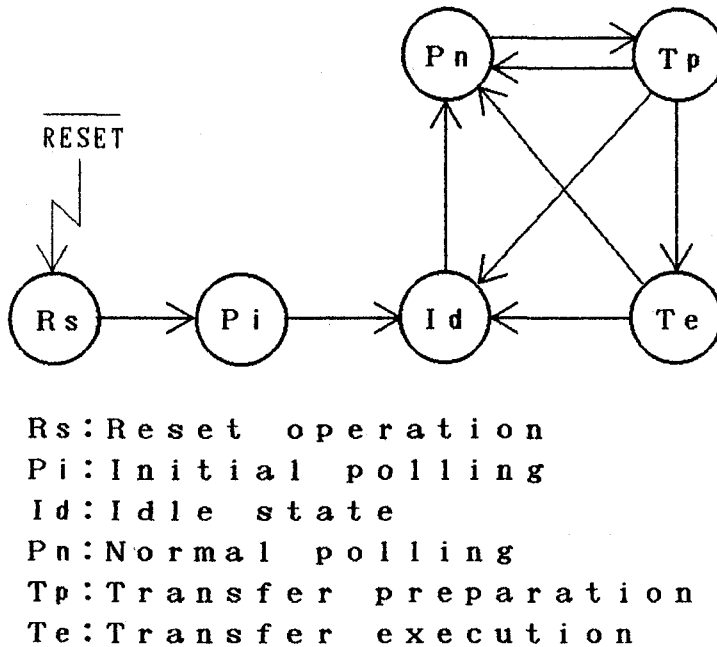


図5-2 LCBの状態遷移

システムリセット後からのLCBの状態遷移を図5-2に示す。Rs、Piの2状態はシステムリセット直後にしか現れない状態なので、初期状態と呼ぶ。初期状態終了後、LCBはId、Pn、Tp、Teの4状態のいずれかに属し、絶えずその間を遷移している。この4状態をまとめて定常状態と呼ぶ。

通常ポーリングでは、複数のCHPUからのバス要求があった場合、バス獲得の優先順位が決まってしまう。しかし、アイドル状態でのみ新たなバス要求を出すことが許され、出されているバス要求に対して、その処理がすべて終了しなければアイドル状態に移行しない。従って、任意のCHPUは有限時間内にLCBを獲得できる。

バスに接続されているUCの台数を n 、転送データ量を s (word)とすると、バス獲得のオーバーヘッドがなかった場合の通信時間 T は次式で表わされる。

$$T = 600 + 100 \cdot L + 40s \text{ (ns)}.$$

但し、 $1 \leq L \leq \lfloor \log_2 n \rfloor$ 。

5. 5 並列処理システムの構成

図5-3, 5-4は、図2-2のLINKS-1の星状あるいは木構造システムと等価なシステムである。LCB及びI/Oプロセッサを介して接続されたデータベースシステム(DBS)上のデータ及びプログラムは、RCに転送される。LCが計算した画素データは、同じくLCB, I/Oプロセッサを介してFMSに出力される。I/Oプロセッサは、汎用プロセッサとCHPを用い、外部装置の種類に応じて作成した。この構成では、LINKS-1で蓄積されたソフトウェア資源をそのまま有効に活用することができる。さらに、LINKS-1では不可能であったNC-NC間、LC-LC間の通信が可能となり、画素間のコヒーレンスの有効利用ができ、メモリ容量不足の解消につながる。

図5-5の配列型構成は、LINKS-1では困難なソフトウェア的な構成の変更や制御が容易であり、故障にも対処しやすい。この場合、従来と同様な並列処理機構は、ソフトウェア的に星状あるいは木構造型の通信経路を設定することで実現できる。ただし、図5-3, 5-4の場合に比べて、通信経路の長さが増加するという欠点がある。また、縦方向の各UCに負荷分散すると同時に、横方向のUCに機能分散を行い、並列パイプライン処理が可能である。さらに、負荷分散手法に関しても、画面分割法だけでなく、空間分割法の実現も容易である等、柔軟性が高い。

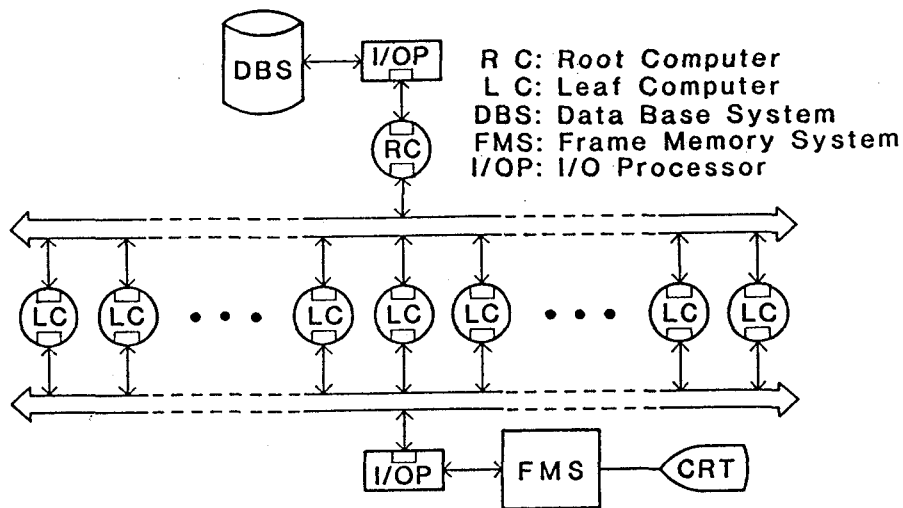


図 5-3 星状構造システム

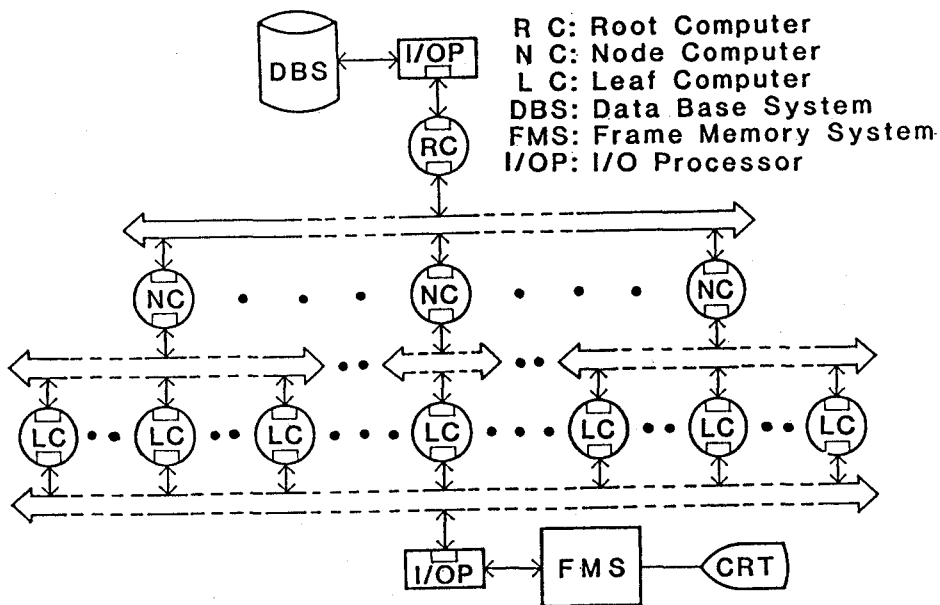


図 5-4 木構造システム

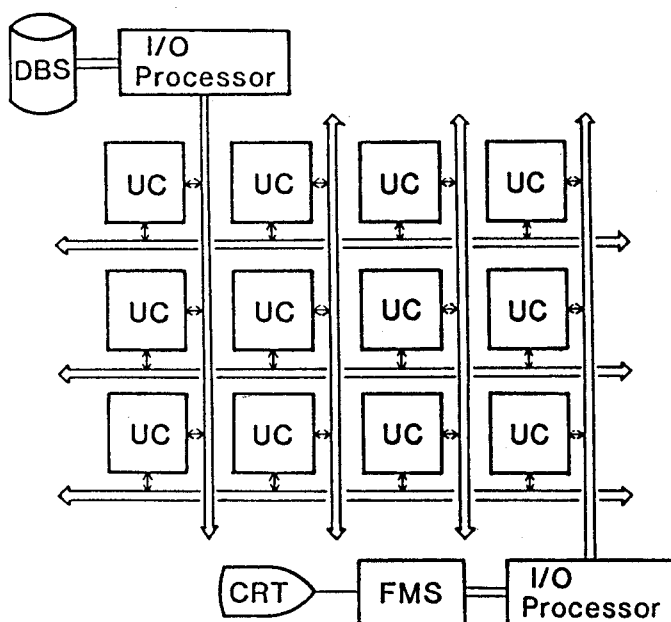


図 5-5 配列構造システム

5. 6 通信処理方式の評価

演算処理能力に対する通信処理能力を評価するため、UCを複数台並列に動作させたときの処理速度の向上比（並列処理効率）をシミュレーションによって求めた。ここでは、総LC台数がNであるようなシステムによる画像生成時間を $T(N)$ と表わし、並列処理効率（Performance）を

$$P(N) = (1/T(N))/(1/T(1)) = T(1)/T(N)$$

と定義する。

図 5-6 に示す球 99 個からなる画像を、デマンド転送方式により生成した場合の画像生成時間、及び並列処理効率をLINKS-1での場合と比較する。システム構成は図 5-3 に示した星状構造とする。実験時の条件は、反射・透過なし、全画素数が 512×512 、1回に割当てられる小画面中の画素数を 4×4 、 8×8 、 16×16 、 32×32 画素の 4 種類とした。

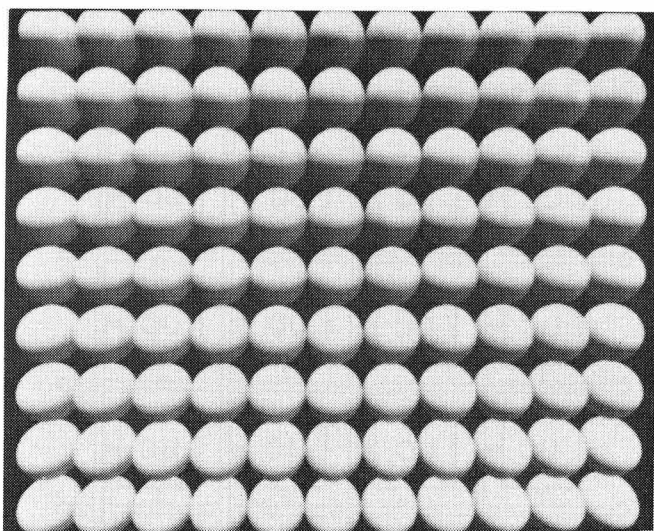


図 5-6 実験に用いた画像

LINKS-1では、LC台数を変えながら、実際に画像を生成し、その所要時間を測定した。LINKS-2の場合は、LINKS-1と同様な実行制御を行うため、まず画像生成時のLCBを用いたRC-LC間の通信を次のようにモデル化した。

・RCの処理：

- (1) LCからの割込みを待つ。
- (2) LCからの小画面要求があれば、CHPUを起動、小画面データ(20バイト)を転送。
- (3) (1)に戻る。

・LCの処理：

- (1) CHPUを起動し、バスを獲得する。
- (2) RCに小画面を要求する。
- (3) 小画面データを受領し、バスを解放する。

(4) もし物体データが必要なら, CHPUを起動, バスを獲得する.

(5) バス獲得後, RCの内部処理に影響を与えることなく, RCのデータメモリより必要な物体データを読み出す. これは, LC側でRCにおける物体データのあるアドレスが分っているためである.

(6) バスを解放し, 小画面の処理を行う.

(7) (1)に戻る.

・マスタCHPUの処理:

5. 4に述べた手順でバスの調停を行う.

以上のモデルを, 計算機上に構築した. CHPUの起動にはコマンドレジスタの設定等で $20\mu s$ かかる. LCがRCから読み出す物体データ量と, LINKS-2における小画面の処理時間が必要であるが, 画像生成プログラム全てをシミュレータで実行することは, 所要時間の点で非現実的である. そこで, まず画像生成プログラムのうち, 交差判定及び輝度計算モジュールの実行時間を, 4. 5で用いたUCシミュレータにより測定した. これらはUCでの処理時間に大きく寄与すると予想される. 次にこれらの実行時間と画像生成プログラムの実行結果から, それぞれの分布を調べた. 表5-2にその平均と分散を示す. このデータから本実験では, 小画面の処理時間を正規分布, 転送データ量を指数分布に従う乱数で近似した.

LINKS-1, 2におけるUC1台の時の画像生成時間を表5-3に, 並列処理効率を図5-7, 5-8に示す. この結果から, LINKS-1では小画面の画素数によって並列処理効率が大幅に異なり, 128台でも高々94台分の能力しかない. これに対しLINKS-2では, 並列処理効率 $P(N)$ は画素数によらずほぼ N に等しく, 最善の結果を得た.

LINKS-1におけるこれまでの実験では, 交差判定並びに輝度計算は, 全処理時間のうち約60~70%であり[NISD84], 実際のUC1台の画像生成速度はLINKS-1のおよそ50倍程度であると考えられる. その場合でも, 並列処理効率は処理時間 t と通信時間 τ の比で決まるため[NISH85b], 図5-

8に示す値を下回ることはない。

総LC台数 N が臨界値 $((\tau+t)/\tau)$ を超えると、並列処理効率 $P(N)$ は、 N に関係ない一定値に収束する[NISH85b]。表5-2の平均処理時間、平均データ転送量から臨界値を見積もると、小画面の画素数によって異なるが、 8×8 画素のときおよそ2800台となった。この値は大規模なシステム構築の有効性を示唆する。

以上の結果を総合して、演算速度の向上に見合うだけの通信処理速度の改善が実現された。

表5-2 小画面の処理時間と転送データ量の平均と分散

小画面 (pixels)	処理時間		転送データ量	
	平均(msec)	分散(msec ²)	平均(byte)	分散(byte ²)
4×4	3.84	1.22	472.5	5.773×10^4
8×8	14.99	16.57	468.2	5.586×10^4
16×16	59.53	181.46	560.6	6.865×10^4
32×32	273.23	1645.83	675.0	6.637×10^4

表5-3 LC1台の時の画像生成時間

小画面(pixels)	LINKS-1(sec)	LINKS-2(sec)	速度比
4×4	5716	63.97	89.4
8×8	5051	61.65	81.9
16×16	4807	61.03	78.8
32×32	4715	60.76	77.6

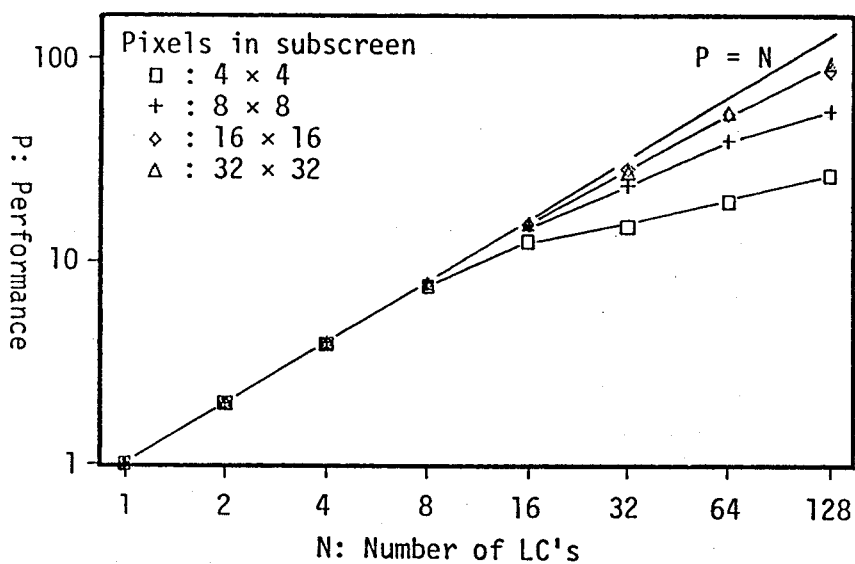


図5-7 LINKS-1の並列処理効率

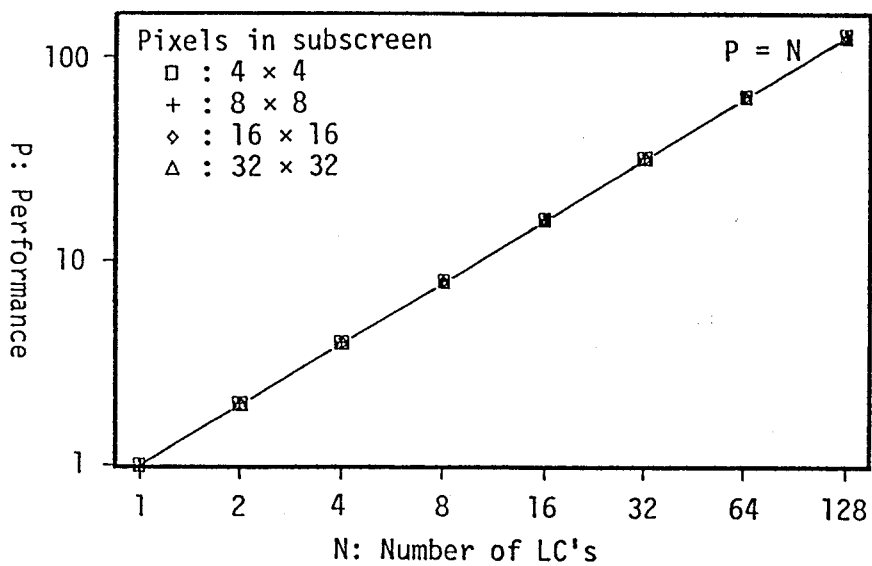


図5-8 LINKS-2の並列処理効率

5. 7 結言

本章では、並列画像生成システムの構築に必要なUC間通信方式について述べた。

本UCは、2組の通信処理専用プロセッサCHPを持ち、2系統の高速ブロック転送バスLCBを介してUC間通信を演算処理と並行して行うことができる。LCBは共有バスであり、バスに接続された任意のUC間の通信が可能である。これにより、システム構成の柔軟性が増し、様々な並列処理方式の実現が可能となった。LCBの調停は、バスに接続された任意のCHPUがこれを行うため、耐故障性に優れている。また、通信誤り検出時の再送機能、動的な通信速度の変更等により、通信の信頼性も確保されている。

次に、シミュレーションにより画像生成時の並列処理効率を求め、LINKS-1の場合と比較した。その結果、画像生成速度、並列処理効率とも改善されており、1000台程度の大規模な並列処理システムの実現の可能性が確認できた。

第6章 空間分割法による

並列視線探索法

6.1 緒言

画面分割法に伴うメモリの使用効率が悪化する、あるいは全物体データを持つ特定のプロセッサに通信が集中しプロセッサ数に見合う性能が得られなくなる等の問題点を解決する方法として、空間分割法[CLEA83][DIPP84]がある。空間分割法は、計算機内に構築された3次元空間をいくつかの部分空間に分割し、視線を各部分空間に進行させながら、それぞれの部分空間で視線探索法を実行する方式である。この方式は、2次元、若しくは3次元格子状に結合されたマルチプロセッサシステム上で実現され、通信は隣接したプロセッサ間のみで行われる。このため、特定のプロセッサに通信が集中することが避けられる。さらに、物体データは分割して各プロセッサが持つため、メモリ効率の改善が図れる。このような特徴は、大規模な並列処理システムにおいて、その規模に見合う性能を引き出すことを可能とする。その一方、空間の分割方式や、各プロセッサにおける負荷の均一化等の問題が新たに生じる。

空間分割方式に関しては、直方体領域に分割する方法[CLEA83][DIPP84]や、Octreeを用いる方法[MATS83][GLAS84]が提案されている。Octreeの場合、その部分空間の管理が複雑となる欠点がある。

負荷分散に関しては、1枚の画像生成の処理中に分割を動的に変化させる方法[DIPP84]や、プロセッサ数に比べて十分多くの均一な部分空間に分割し、そのいくつかを1つのプロセッサに割当て方法[KUBT87]等が提案されている。前者には、変更の方法やタイミング、物体データ転送のオーバーヘッド等難しい問題がある。後者には、分割数の増加により、各プロセッサが重複して管理しなければならない境界領域に存在する物体データが増え、メモリ使用効率が悪化する等の問題がある。

既に画面分割法において、1次視線に対する様々な高速化手法は大きな効果があることが判っている[DEGU84b]。本論文では、これらを有効に利用す

るため、空間を角錐状に分割する方式を提案する。また、我々は動画制作を主目的としているため、負荷の評価にフレーム間コヒーレンスが利用できる。すなわち、前のフレームの負荷の状況から次のフレームの空間分割を行う。このような手法により、単純でしかも効率的な並列処理が可能となる。本章では、その概要を述べ、シミュレーションによりその効果を明らかにする。

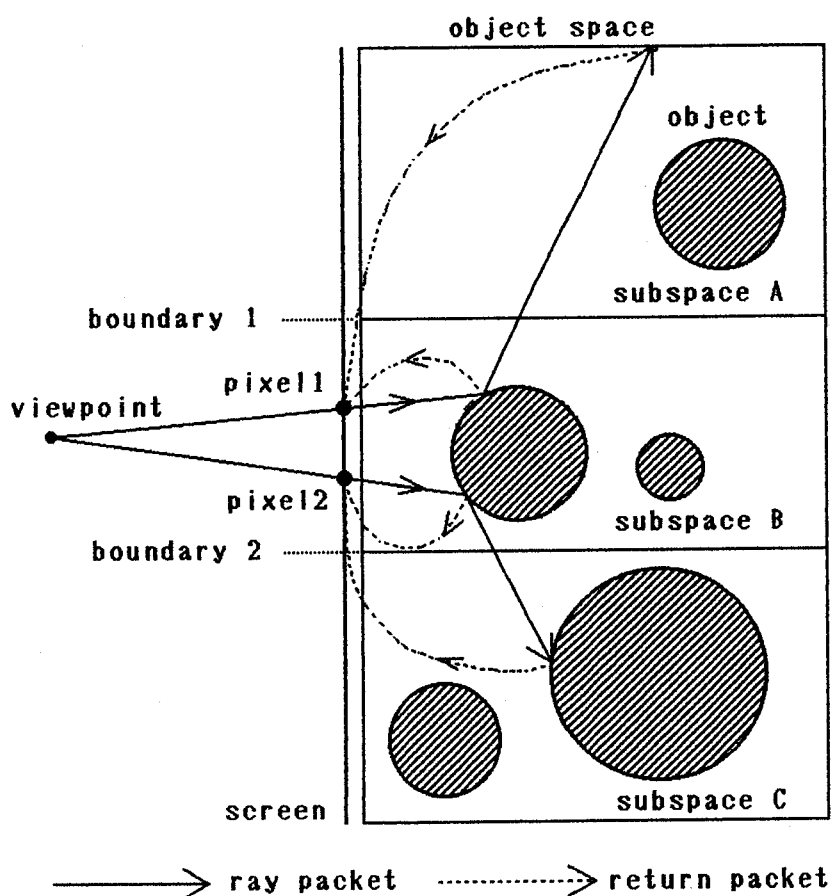


図6-1 空間分割による視線探索法

6. 2 並列化手法

空間分割法は、前述したように、計算機内に構築した3次元空間を複数個に分割し、それぞれの部分空間を、分割空間と同一トポロジを持つマルチプロセッサシステムの1つのプロセッサに割当てて、各プロセッサは、隣接した部分空間を担当するプロセッサと通信ハードウェアにより結合されている。また、各プロセッサは、その部分空間に含まれる総ての物体データ、および部分空間の境界情報を持つ。

1本の視線は、その方向、通過した画素のスクリーン座標、その画素に対する輝度の貢献度などを持つ1つのパッケージ（視線パッケージ）として表わされる。

各プロセッサは、視線パッケージ毎に、部分空間内の物体との交差判定を行う。交差しなければ、その視線パッケージは、視線の進行方向にある隣接したプロセッサに渡される。交差する場合、その点での散乱反射による輝度を計算し、リターンパッケージとしてフレームメモリに転送する。付影処理をするときは、この時点ではリターンパッケージを作らず、各光源方向に新たな視線パッケージを送り、光源に達した時点でその光源による輝度のリターンパッケージを生成する。さらに反射・屈折方向に新たな視線パッケージを送る。視線パッケージが物体の存在領域外に出たならば、背景の輝度を求めリターンパッケージを生成し、その視線についての処理を終わる。視線パッケージが無くなれば、全処理が終了したことになる（図6-1）。

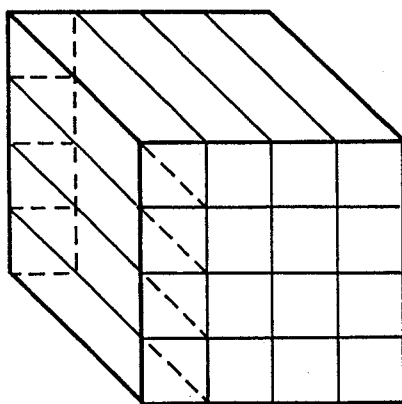
各プロセッサの負荷は、 P_s をそのプロセッサの処理する視線数、 B_s を部分空間内に存在する物体数、 $f(n)$ を物体数 n のときの1本の視線に対する交差判定の手数とすると、 $P_s \cdot f(B_s)$ に比例する。また、各部分空間を通過する視線の数は、その空間の境界の面積、すなわち部分空間の表面積に比例すると考えられる。従って、分割数を N としたときの実行時間は並列処理を行わない場合に比べて、 $(S_N/S) \cdot (f(B_N)/f(B))$ 倍となる。但し S 、 B は、分割前の全物体を囲む表面積と全物体数であり、 S_N 、 B_N は N 分割後の部分空間の表面積と物体数である。

Dippeらは、空間分割による部分空間内の物体数の減少を積極的に評価し、

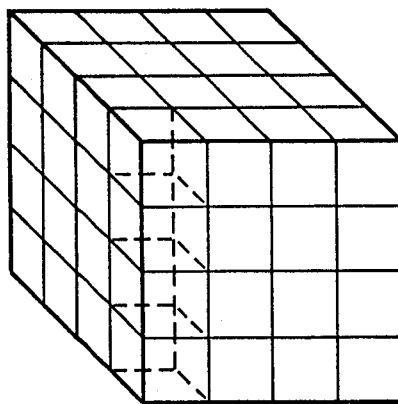
結果として $f(n)$ が物体数に比例すると仮定した[DIPP84]。しかし、一般に物体データの階層化等による交差判定の高速化手法[DEGU84b] が取り入れられているため、 $f(n)$ は物体数の対数に比例するか、あるいは定数となる[RUBI80]。それ故、分割数の増加による実行時間の変化は、主として部分空間の表面積の変化によるものとなる。

6. 3 空間分割手法

空間の分割手法は、後述する負荷分散手法とも関連し、並列化において重要な問題の1つであり、システムの構成にも多大な影響がある。Octreeを用いた方法は、基本的にシングルプロセッサ向き的高速化手法であり、部分空間の隣接情報などの管理が困難である。



(a) 2D mesh subdivision



(b) 3D mesh subdivision

図6-2 2次元及び3次元格子状の空間分割

2次元あるいは3次元格子状の直方体領域に分割する方法[CLEA83](図6-2)は、単純なシステム構成で実現可能であり、マルチプロセッサ向きである。N分割したときの各部分空間の表面積は、それぞれ($aN^{-1} + bN^{-1/2}$)、および $cN^{-2/3}$ となる。但し、aは分割前の空間の表面積cのうち、画面を含む境界とそれに向かい合う境界の面積の和であり、bはそれ以外の境界の面積の和である。従って、2次元格子状に空間をN分割した時、

(1) 2次視線が殆ど発生しない場合はNに比例した、

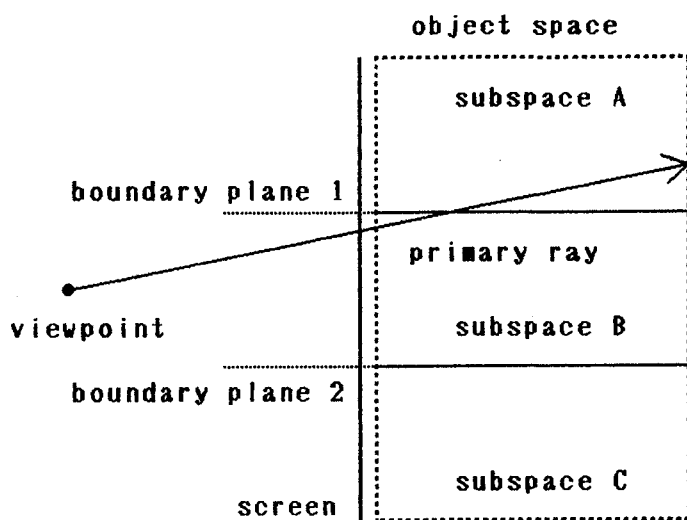
(2) 1次視線に比べて十分に多い2次視線が発生する場合は $N^{1/2}$ に比例した、

実行速度が期待でき、3次元格子状に分割した時は $N^{2/3}$ に比例した実行速度が期待できる。

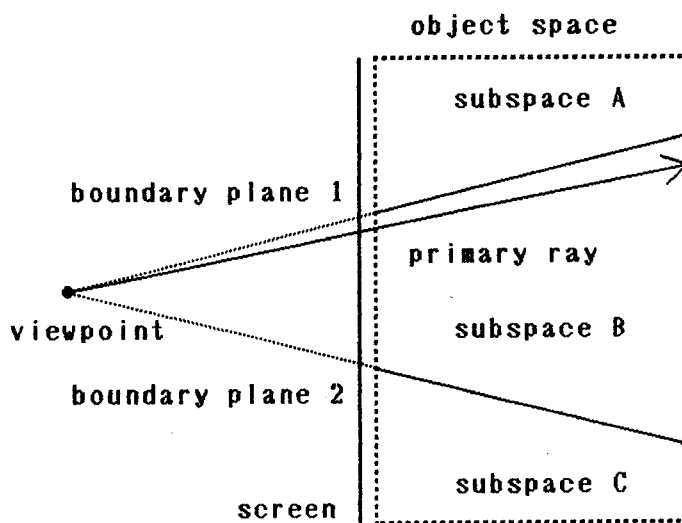
3次元格子状に分割する場合、プロセッサ数の増加による部分空間の表面積の減少が、2次元格子状に分割する場合に比べて大きい。そのため、大量の2次視線が発生する場合は、プロセッサ数の増加による性能向上の点において有利である。しかし、システムの構成が3次元格子状となり、プロセッサ間の接続が複雑になる。

一方、2次元格子状に分割する場合、スクリーン上でのクリッピング等、1次視線処理の高速化手法[DEGU84b]を適用することにより、交差判定しなければならない物体数の効果的な削減が可能である。この手法は、各物体の外接直方体等をスクリーン上に投影し、その投影像を囲む長方形領域を予め求めておき、一次視線と交差する物体の抽出に役立てるものである。この手法は整数演算のみで実行でき、大きな効果を上げることがわかっている。従って2次視線数がそれほど多くない場合は、3次元格子状に分割するより有利であると考えられる。

以上の理由から、我々は空間の分割手法として、より単純なシステム構成で実現可能な2次元格子状に分割する方法を採用した。また、制御の容易さから、分割にはスクリーン座標軸に平行な平面(分割平面と呼ぶ)のみを用いることにした。



(a) Space subdivision by parallel planes.



(b) Space subdivision by planes through viewpoint.

図6-3 空間分割方式（平行分割と角錐分割）

2次元格子状に分割の際、スクリーンに垂直な平面を用いた場合（平行分割方式と呼ぶ）、図6-3（a）に示したように1次視線が複数の部分空間を通過することがある。そこで、図6-3（b）に示すように必ず視点を通過する平面を用いて空間を分割する（角錐分割方式と呼ぶ）。この方式では1次視線は1つの部分空間しか通過しないため、各部分空間を担当するプロセッサにおいて、1次視線に対する処理を一括して行うことができる。全く2次視線が発生しない場合は、画面分割法にほぼ一致する。よってプロセッサ数に比例した処理速度の改善が得られる。

6. 4 負荷分散手法

空間分割を用いて視線探索法を並列化する際、なるべく各プロセッサにおける処理時間が均一になるように部分空間を割当てて必要がある。Dippeらは、1枚の画像生成中に、ある部分空間の負荷が重いと判明した場合、その部分空間の頂点を局所的に移動することで、周囲の部分空間にその空間の一部を内部の物体データと共に分け与えることで負荷の均一化を図ろうとした[DIPP84]。しかし、その判定の基準やタイミング、分割形状の変更アルゴリズム、物体データ転送のオーバーヘッド等困難な問題が多い。

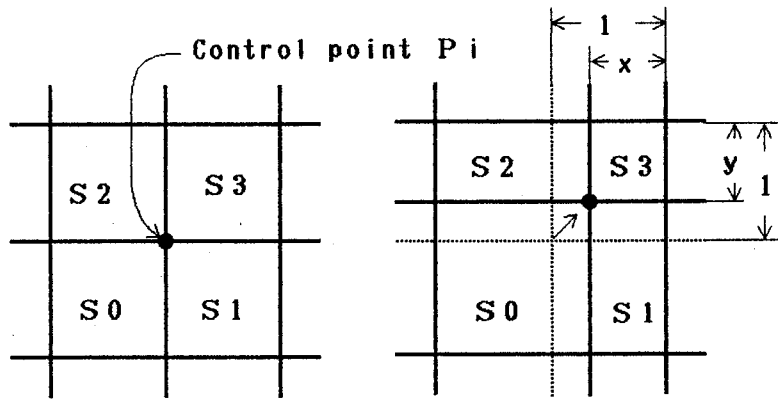
実際の画像創作活動では、同じ画像の生成は1回だけ行われるのではなく、解像度等を変え、何回か繰り返し行われることが多い。あるいは、動画作成においては連続的な画像生成が行われ、隣接画像の間には類似性（フレーム間コヒーレンス）が見られる。従って、準備的な画像生成における処理時間や、動画においては1フレーム前の画像の処理時間を負荷分布の類推に利用できる。そこで我々は、画像生成中の動的な負荷の変更を行わず、与えられた負荷の状況に対してより均等な負荷分散を行うよう分割しなおす方法を採用した。すなわち、以前に分割および画像生成を行った際の各部分空間の処理時間を測定しておき、このデータにより分割を変更する。この手法を適応再分割法と呼ぶ。

この手法を適用するにあたり、次の仮定が成り立つものとした。

（1）6. 3で述べた空間分割方式を用いる。これらの方式における最

適な負荷分散を目指す。

(2) 各部分空間での負荷は、空間中で一様に分布しており、ある部分空間の境界のうちスクリーンを含む境界(=小画面)の一部を他の部分空間に移動することにより、その面積に比例した負荷が移動されるものとする。すなわち、1次視線数および物体数の調節により負荷の均等化が可能であるとする。



(a) initial division (b) after redivision

図6-4 負荷の移動

画面上で水平と垂直方向の分割平面の交わる点を制御点と呼ぶ。制御点の移動により部分空間の負荷の大きさを調整する。

図6-4 (a)において、制御点Piの周囲の4つの小画面Siのうち、対応する部分空間の負荷 L_i が最も大きいものをS3とし、その負荷を L_3 とする。 $L_0 \cdot L_3 \leq L_1 \cdot L_2$ であるとき、S3における負荷を減らし、

$$L_1' = L_2' = L_3' \geq L_0',$$

とすることができる(図6-4 (b))。このときの x , y は、

$$x = (L_2 + L_3) / 2 L_3,$$

$$y = (L_1 + L_3) / 2 L_3,$$

である。 $L_0 \cdot L_3 \leq L_1 \cdot L_2$ の条件を満たさない場合、最小の負荷をもつ部分空間の負荷を大きくすることで均一化を図る。 S_0 が負荷最小とすると、

$$L_0' = L_1' = L_2',$$

とするには、

$$x = (3ab - (9a^2b^2 - 8abcL_3)^{1/2}) / 4aL_3,$$

$$y = (a/b)x,$$

$$\text{但し, } a = L_1 + L_3, \quad b = L_2 + L_3,$$

$$c = L_0 + L_1 + L_2 + L_3.$$

小画面 S_3 の幅を w 、高さを h とすると、制御点 P_i の移動ベクトルは、

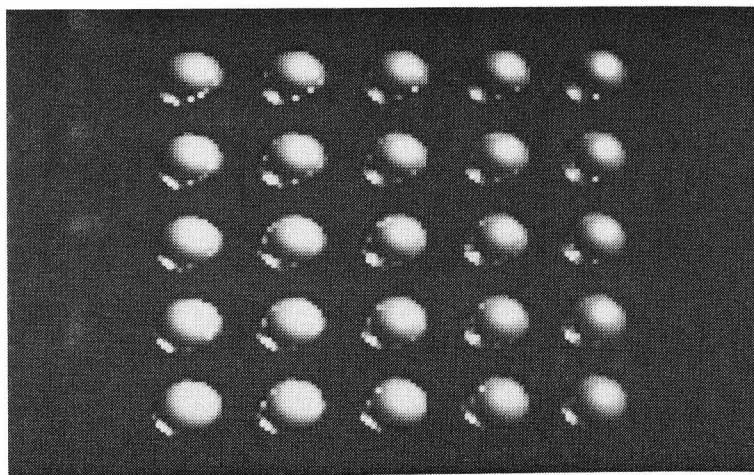
$$((1-x)w, (1-y)h).$$

上式により、すべての制御点の移動ベクトルを求め、各分割平面上の制御点の移動ベクトルの和をその平面の移動量とする。

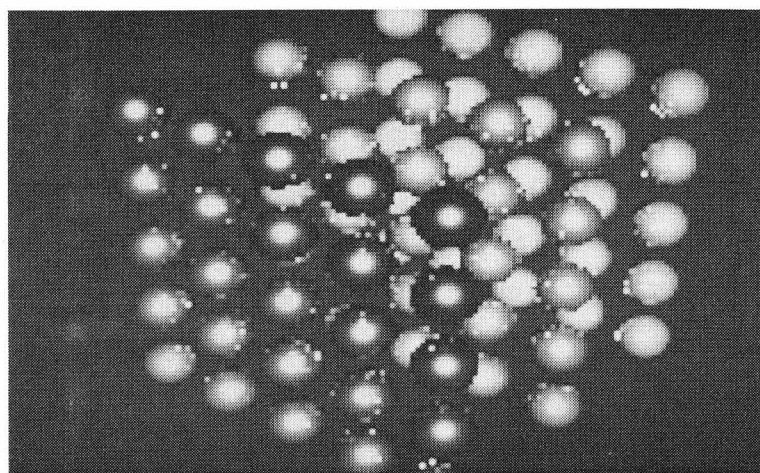
6. 5 シミュレーションによる評価

本章で述べた手法による並列視線探索法を、マルチタスクOSの稼働している計算機上で、各プロセッサをそれぞれ1つのプロセスとして、プロセッサ間通信をプロセス間通信に置き換えて実現した。さらに、図6-5に示す画像のデータを用いて、負荷分散の効果、並列処理効率、メモリ使用量について実験を行った。図6-5(a)の画像例1では楕円体を25個、(b)の画像例2では楕円体75個を使用し、反射回数は2回とした。

画像例1のデータを使用し、分割数を9とした場合の適応再分割による画像生成時間の変化を、図6-6に示す。この実験例では、4回目の再分割で画像生成時間が最小となり、その時の分割の様子を図6-7に示す。この結果から、適応再分割を繰り返すことで、画像生成時間はある一定値に収束していくことがわかる。但し、その収束の速さはデータによって異なる。2次視線の影響が大きい場合や物体が局所的に集中している場合は、6.4で述べた(2)の仮定が成立しなくなるため、収束に時間がかかる。



(a) 画像例 1



(b) 画像例 2

図 6-5 実験に用いた画像

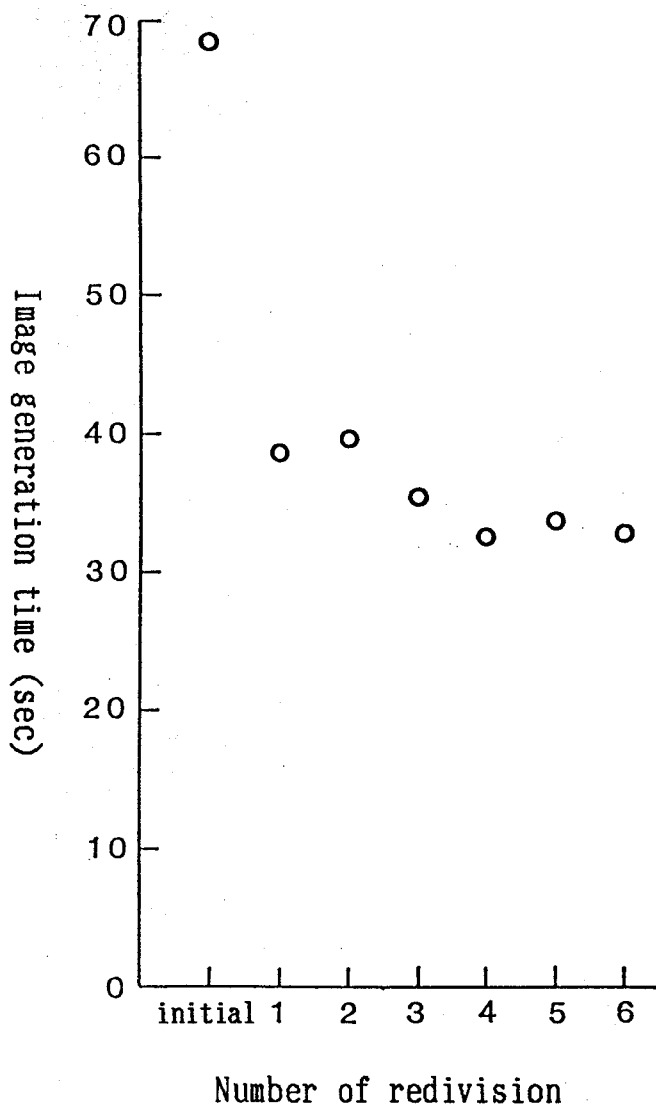
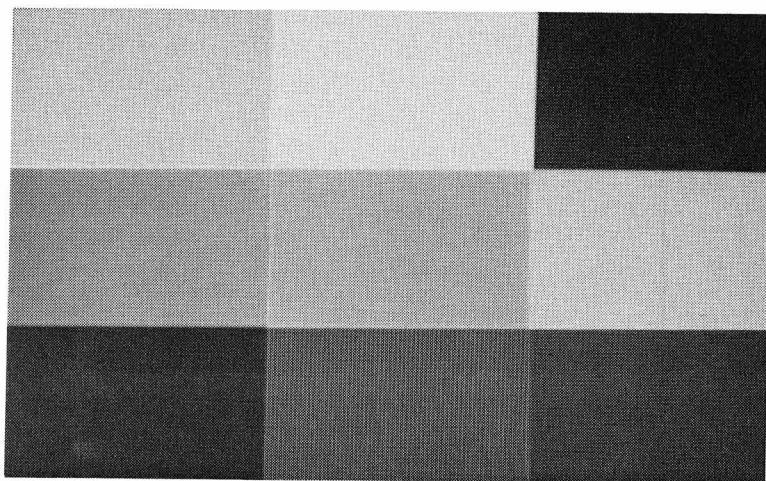
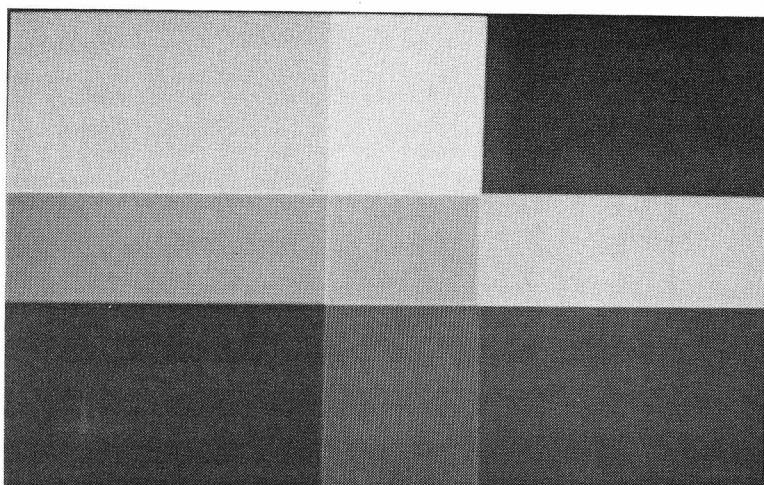


図6-6 負荷分散の効果



(a) 初期分割



(b) 負荷分散完了時

図6-7 初期分割と負荷分散完了時の分割

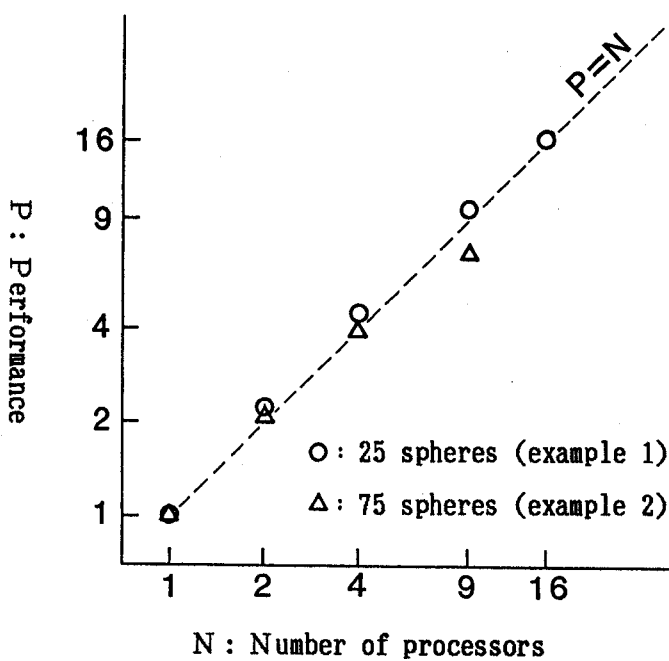


図6-8 空間分割法を用いた時の並列処理効率

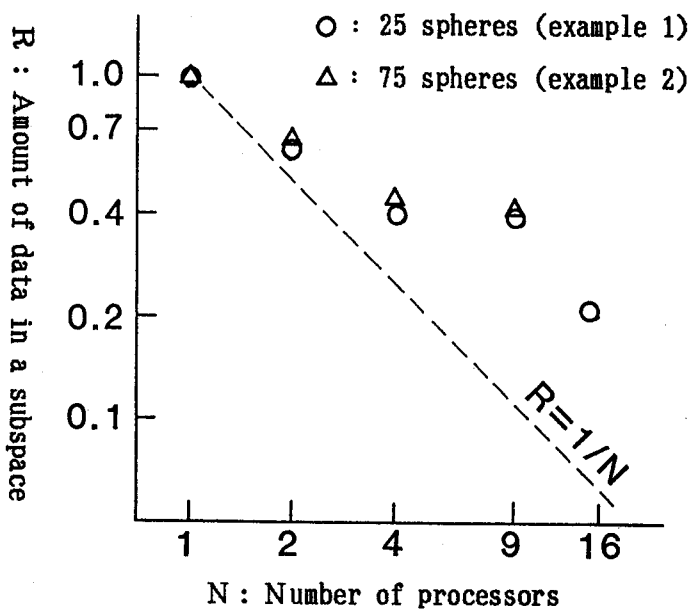
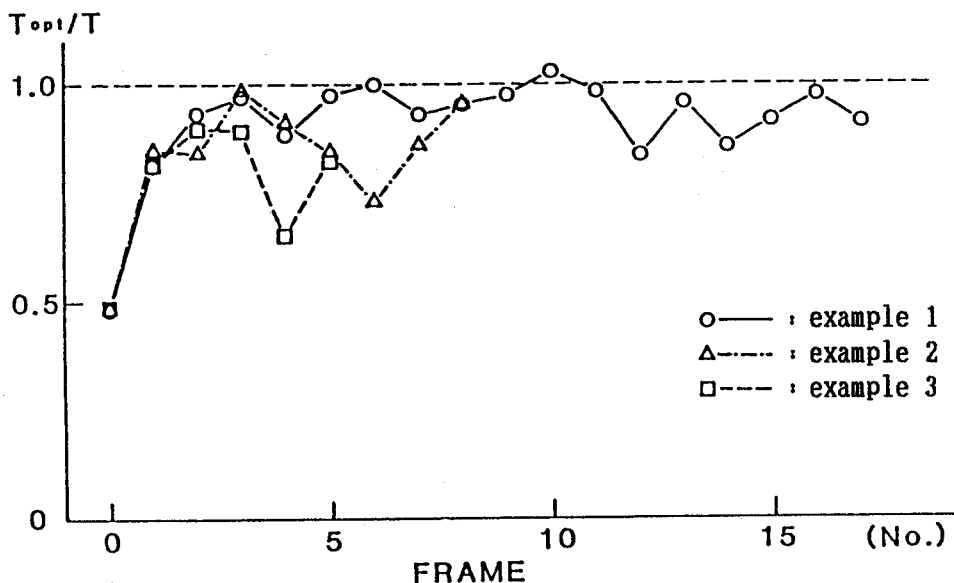


図6-9 部分空間内の物体データ量の変化

次に、分割数（プロセッサ数）を変えた場合の並列処理効率を図6-8に示す。この結果、ほぼプロセッサ数に比例して画像生成速度が向上することがわかった。特に画像例1では、

- (1) 2次視線の数が非常に少ないこと、
 - (2) 物体データの階層化が十分に行われていないため、物体数の減少による交差判定の高速化の効果が顕著であること、
- から、プロセッサ数に比例する以上の高速化が達成されている。

さらに、分割数による1プロセッサ当りのメモリ使用量の変化を調べた。その結果を図6-9に示す。この図から、分割数の増加に伴い減少していることがわかる。分割数が大きいとき減少率が低いのは、部分空間の境界付近の物体データを隣接するプロセッサが重複して持っているためである。



T_{opt}: Optimal image generation time for 6 times of redivision
T : Image generation time of a frame in making animation

図6-10 動画作成時の負荷分散の効果

動画においては、前述したように、1フレーム前の分割とその各部分空間での負荷の状況を利用して適応再分割を行うことができる。そこで次のような実験を行った。物体データとして画像例1と同じものを使用し、このデータ全体を水平軸まわりに回転させた。回転速度は、動画例1が $10^\circ/\text{frame}$ 、動画例2が $20^\circ/\text{frame}$ 、動画例3が $30^\circ/\text{frame}$ 、分割数は12、反射回数は2回とした。その結果を図6-10に示す。この図から、動画生成に適応再分割を用いることが有効であることがわかる。

6. 6 結言

本章では、空間分割法による並列視線探索法について述べた。本手法の特徴は、以下の通りである。

(1) プロセッサ数を N とすると、2次視線が大量に発生する場合は $N^{1/2}$ に、2次視線が少ない場合は N に比例した並列処理効率が得られる。

(2) 1プロセッサ当りのメモリ使用量は、分割数の増加に伴い減少する。1枚の画像生成に使用できる物体データ量は、複数の部分空間にわたって存在する物体データの全物体データに対する割合を δ とすると、画面分割法のバッチ転送方式に比べて、 $(N/(1+\delta))$ 倍となる。

空間分割手法としては、1次視線処理の高速化の容易さから角錐分割を、負荷分散には、前回の各部分空間における画像生成時間を利用した適応再分割法を提案した。また、シミュレーションによりこれらの手法の有効性を明らかにした。

第7章 結 論

本研究で得られた成果について簡単にまとめておく。

第2章では、マルチコンピュータシステムLINKS-1の概要並びに視線探索法を用いた並列画像生成手法について述べた。さらに、LINKS-1での様々な実験結果から、演算処理能力、UC間の通信処理能力の不足が画像生成の高速化の妨げとなっていることを示した。

第3章では、LINKS-1での実験に基づき、新たに設計した並列画像生成システムLINKS-2のユニットコンピュータの概要について述べた。演算の高速化を図るためには、UC内部のデータ転送をできる限り高速に行うことが重要であることに着目して設計を行った。本研究では、データメモリとプログラムメモリの分離、2項演算を1命令で実行するマイクロプログラム方式、アドレス演算専用処理ユニット、データメモリのインタリーブ構成等の手法を用いた。乗算器については、論理機能素子の遅延時間に基づき計算機による自動設計を行った。

第4章では、LINKS-2での演算処理方式について述べた。4ウェイインタリーブ構成のデータメモリを、CBSを介して3組のデータバスに接続することで演算データの転送を並列化した。UC内部の各ユニットは、2項演算を並列パイプライン処理する。そのモデルを計算機上に構築し、データメモリへのアクセス競合がパイプライン処理に及ぼす影響を調べた。その結果、画像生成処理においては、その影響は十分小さいことがわかった。

第5章では、演算処理と並行して動作する通信専用プロセッサ、32ビット高速ブロック転送バスを用いた通信機能の強化手法について述べ、シミュレーションによりその効果を明らかにした。演算速度の改善だけでは、並列処理時の効率が低下するため、通信速度の改善が同時に必要となる。シミュレーションの結果、LINKS-2では、それぞれの改善がバランス良く行われていることがわかった。LCが100台程度のシステムでもそれに見合う処理能力の向上が期待でき、さらには、設計上の目標であるLC1000台程度の大規模なシステムの構築に関する指針が得られた。システム完成後は、高

品質な画像を十分高速に生成できるものと考えられる。

第6章では、空間分割法による視線探索法の並列化について述べ、シミュレーションによる評価を行った。本研究では、角錐型空間分割法、適応再分割による負荷分散法を用いた。この結果、1プロセッサ当りの必要メモリ量は、複数の部分空間に渡って存在する物体データの全物体データに対する割合を δ とすると、画面分割法（バッチ転送方式）の場合に比べて、 $(1 + \delta) / (\text{プロセッサ数})$ に削減でき、並列処理効率は、ほぼプロセッサ数に比例することが確認できた。この結果から、大量の物体データから構成されるシーンを十分高速に生成する並列処理方式として空間分割法が有効であることがわかった。

最後に、残された課題についてまとめておく。

(1) LINKS-2の性能を十分に発揮させるためには、最適化されたコードを出力するコンパイラを始めとするソフトウェアの充実が必須である。

(2) 本論文でのシミュレーションは、LINKS-1と同様な並列画像生成処理を仮定しており、演算処理と通信処理とを分離して扱うなどごく限られた条件のもとで行ったにすぎない。LINKS-2は、演算と通信を並行して処理できるという特徴を持ち、LINKS-1にはないシステム構成の柔軟性に富んでいる。よってさらに多様な条件や、構成のシステムでのふるまいを明らかにする必要がある。

(3) 空間分割法による並列視線探索法に関しては、部分空間内における物体データを計算に適した構造に再構成する手法や、適応再分割を行うためのよりよい初期分割法を考案する必要がある。

(4) 映像表現の一手段としてのCGの定着に伴い、シーンを構成するデータは、ますます複雑化し、その量は増加の一途を辿っている。LINKS-2の実用化に当たっては、大量のデータから構成されるシーンの高速な画像生成が要求される。そのためには、空間分割法の適用を含めたLINKS-2に適したシステム構成法、並びに並列処理方式の検討が必要である。

謝 辞

本研究の全過程を通じて、終始懇切な御指導、御助言を賜った寺田浩詔教授に衷心より謝意を表する。

終始一貫して御指導、御教示を賜った大村皓一助教授（現在、大阪学院大学教授）に心から謝意を表する。

日頃から温かい御支援、御助言を頂いた寺田研究室の浅田勝彦助教授（現在、福井大学教授）、笹尾勤助手（現在、九州工業大学助教授）、西川博昭助手、江木康雄技官に深く感謝する。

本研究を進めるにあたり、種々の面で御指導、御助言を賜わった尾崎弘名誉教授、白川功教授、河田亨助教授（現在、シャープ^(株)）、築山修治助教授（現在、中央大学助教授）、藤原秀雄助手（現在、明治大学助教授）、出口弘助手に深謝する。

大学院前期、後期両課程を通じて電子工学一般および各専門分野に関し御指導、御教示賜わった電子工学教室の小山次郎名誉教授、西原浩教授、浜口智尋教授、児玉慎三教授、電子ビーム研究施設の裏克己教授、埴輝雄教授、産業科学研究所の松尾幸人名誉教授、中村勝吾教授（現在、熊本電波工業高等専門学校校長）、角所収教授、ならびに故中井順吉名誉教授に感謝する。

本研究に関し、常に御助言、御協力頂いた、基礎工学部情報工学教室の宮原秀夫教授ならびに下條真司助手に心から謝意を表する。

本研究においてシステム作成ならびに評価実験等に御協力頂いた、本学卒業生、^(株)西村技研の西村仁志氏、^(株)イマジカの大野廣司氏、吉村浩氏、田中伸治氏、井川勲氏、山下伸一氏、シャープ^(株)の吉田茂氏、岸元泰親氏、舟渡信彦氏、若井裕久氏、松下電器産業^(株)の鷺島敬之氏、^(株)木泰治氏、西村明夫氏、日高教行氏、平井誠氏、辰己敏一氏、水野洋氏、日本電信電話^(株)の高田久靖氏、内藤岳氏、山名岳志氏、正田博司氏、日本電気^(株)の中山貴司氏、内村敏幸氏、那須雅樹氏、西田政人氏、富士通^(株)の高山浩一郎氏、藤田卓志氏、マツダ^(株)の村上毅氏、日本アイ・ビー・エム^(株)の栗山繁氏、富士写真フィルム^(株)の望月康幸氏、奈良工業高等専門学校の世古忠助教授、山井成良助手、

大阪府立工業技術研究所の中西隆氏，古野電気㈱の平岡康氏，近藤仁志氏，
㈱野村総合研究所の高橋良浩氏，和田充弘氏，本田技研工業㈱の澤多靖浩氏，
元研究生の小林弘明氏，高沖英二氏，本学大学院生の井上博之氏，ならびに
関西大学卒業生，三菱電機㈱の岸良行氏に感謝の意を表する。

システム利用者の立場から有益なるご助言を頂いたイメージファクトリー
の松浦季里嬢，中川佳子嬢，岡勝利氏，太陽企画㈱の中澤周一氏，石田晃氏，
佐藤裕英氏，㈱リンクスの福本隆司氏，並木茂氏に感謝する。

本研究を進めるにあたり，有益な御助言ならびに御援助を頂いた㈱リンク
ス，㈱イマジカ，シャープ㈱，松下電器産業㈱，北川情報機器㈱，㈱ジェッ
クならびに日本放送協会の関係各位に感謝する。

寺田研究室秘書の藤原先代嬢，前秘書の三谷典子嬢，西田典子嬢，元秘書
の高田有香子嬢，白川研究室秘書の中尾有子嬢，前秘書の前川礼子嬢には種
々の面で御協力頂いた。ここに記して感謝する。

日頃から御支援頂いた寺田研究室，白川研究室，関西大学電気工学科尾崎
研究室の皆様我心から感謝の意を表する。

参考文献

- APPE68 Appel, A., "Some Techniques for Shading Machine Renderings of Solids", AFIPS SJCC, pp.37-45 (1968).
- ATHE78 Atherton, P., Weiler, K. and Greenberg, D., "Polygon Shadow Generation", Proc. ACM SIGGRAPH'78, 12, 3, pp.275-281 (1978-8).
- BENN75 Bennett, W.S., "Computer Generated Graphics: A Review of Some of the More Well Known Methods Applicable to Simulation", Proc. SPIE, 59, Simulators & Simulation, pp.3-11 (1975).
- BLIN76 Blinn, J.F. and Newel, M.E., "Texture and Reflection in Computer Generated Images", Comm. ACM, 19, 10, pp.542-547 (1976-10).
- BLIN77 Blinn, J.F., "Models of Light Reflection for Computer Synthesized Pictures", Proc. ACM SIGGRAPH'77, 11, 2, pp.237-241 (1977).
- BLIN80 Blinn, J.F., Carpenter, L., Lane, J. and Whitted, T., "Scan Line Methods for Displaying Parametrically Defined Surfaces", Comm. ACM, 23, 1, pp.23-34 (1980-1).
- BOUK70 Bouknight, W.J. and Kelley, K.C., "An Algorithm for Producing Half-tone Computer Graphics Presentations with Shadows and Movable Light Sources", AFIPS SJCC, pp.1-10 (1970).
- BUIT75 Bui-Toung Phong, "Illumination for Computer Generated Pictures", Comm. ACM, 18, 6, pp.311-317 (1975-6).
- CATM75 Catmull, E., "Computer Display of Curved Surfaces", Proc. IEEE Conf. Comput. Gr. Pattern Recogn. Data Structure, p.11 (1975-5).

- CLAR82 Clark, J. H., "The Geometry Engine: A VLSI Geometry System for Graphics", Proc. ACM SIGGRAPH'82, 16, 3, pp.127-133 (1982-7).
- CLEA83 Cleary, J. G., Wyvill, B., Birtwistle, G. M. and Vatti, R., "Multiprocessor Ray Tracing", Res. Rep. No.83/128/17, Dept. of Comput. Sci., The Univ. of Calgary, (1983-10).
- COOK81 Cook, R. L. and Torrance, K., "A Reflectance Model for Computer Graphics", Proc. ACM SIGGRAPH'81, 15, 3, pp.307-316 (1981-8).
- DEGU82 出口, 河合, 中西, 西村, 河田, 白川, 大村, "プログラミングシステム LINKS - C", 情処学マイコン研資, 24-3 (1982-11-2).
- DEGU84a Deguchi, H., Nishimura, H., Yoshimura, H., Kawata, T., Shirakawa, I. and Omura, K., "A Parallel Processing Scheme for Three-Dimensional Image Generation", Proc. ISCAS'84, pp.1285-1288 (1984-5).
- DEGU84b 出口, 西村, 吉村, 河田, 白川, 大村, "コンピュータグラフィックスシステム LINKS - 1 における画像生成の高速化手法", 情処学論, 25, 6, pp.944-952 (1984-11).
- DEGU86 出口, 西田, 西村, 河田, 白川, 大村, "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(D), J69-D, 2, pp.170-179 (1986-2).
- DIPP84 Dippe, M. and Swensen, J., "An Adaptive Subdivision Algorithm and Parallel Architecture for Realistic Image Synthesis", Proc. ACM SIGGRAPH'84, 18, 3, pp.149-158 (1984-7).
- FLYN72 Flynn, M. J., "Some Computer Organizations and Their Effectiveness", IEEE Trans. Comput., C-21, 9, pp.948-960 (1972).
- FUCH80 Fuchs, H., Kedem, Z. M. and Naylor, B. F., "On Visible

- Surface Generation by A Priori Tree Structures", Proc. ACM SIGGRAPH'80, 14, 3, pp.124-133 (1980-7).
- FUCH85 Fuchs, H., Goldfeather, J., Hultquist, J. P., Spach, S., Austin, J. D., Brooks, Jr., F. P., Eyles, J. G. and Poulton, J., "Fast Spheres, Shadows, Textures, Transparencies, and Image Enhancements in Pixel-Planes", Proc. ACM SIGGRAPH'85, 19, 3, pp.111-120 (1985-7).
- FUJI88 藤田, 岸, 栗山, 河合, 大村, "多重節点を持つスプライン基底関数を用いた曲面補間法", テレビジョン学会技術報告, ICS88-16, IPA88-11, (1988-2).
- FUKU84 Fukumoto, T., "Bio-Sensor", ACM SIGGRAPH'84, Film & Video Show (1984).
- FUKU86 福本, 山藤, "イメージスコアとその表現方法について", PIXEL, 50, (1986-11).
- FUNA85 舟渡, 出口, 西田, 西村, 河田, 白川, 大村, "コンピュータグラフィックスシステムLINKS-1における並列処理の性能評価", 信学技報, CAS84-203, pp.95-102 (1985-2-1).
- GLAS84 Glassner, A. S., "Space Subdivision for Fast Ray Tracing", IEEE Comput. Gr. & Appl., pp.15-22 (1984-10).
- HIDA85 日高, 平井, 中瀬, 浅原, 鷺島, "マルチコンピュータ画像生成システムMC-1", 情報学計算機アーキテクチャ研資, 58-5, (1985-6-21).
- HIRA83 平井, 西村, 河田, 白川, 大村, "分布関数による物体表現と効率的画像生成の一手法", テレビジョン学会技報, IPD81-5, pp.21-26 (1983-10-27).
- HIRA84 平井, 中山, 出口, 西村, 河田, 白川, 大村, "木構造画像生成システムのシステムモニタの一作成", 昭59信学総全大, 7, p.35 (1984-3).
- HIRA86 平井, 日高, 浅原, 鷺島, "画像生成用SIMD型マルチプロセッサ

- システムMC-2", 情処学マルチメディア通信と分散処理研資,
29-5 (1986-5-15).
- ICHI79 市田, 吉本, "シリーズ新しい応用の数学20: スプライン関数とその
応用", 教育出版 (1979).
- ISHI88 石井, 池坂, 石畑, "高並列計算機CAP", 信学論(D), J71-D,
8, pp.1375-1390 (1988-8).
- KAJI82 Kajiya, J. T., "Ray Tracing Parametric Patches", Proc. ACM
SIGGRAPH'82, 16, 3, pp.245-254 (1982-7).
- KAWA82 河合, 吉村, 出口, 西村, 河田, 白川, 大村, "画像データ操作シ
ステムLINKS-DMS", 情処学マイコン研資, 24-5(1982-11-2).
- KAWA84 河合, 西村, 出口, 河田, 白川, 大村, "LINKS-1によるアニ
メーションの製作について", 情処学グラフィックスとCAD研資,
14-4 (1984-9-4).
- KAWA85a 河合, 若井, 正田, 西村, 近藤, 高山, 出口, 白川, 大村, "画像生
成用マルチコンピュータシステムとプロセッサについて", 情処学グ
ラフィックスとCAD研資, 19-4 (1985-10).
- KAWA85b 河合, 木本, 那須, 河田, 白川, 大村, "コンピュータグラフィッ
クス用画像データベースシステム", NICOGRAPH'85論文集,
pp.21-29 (1985-11).
- KAWA86a 河合, 大村, "LINKS-2", NICOGRAPH'86論文集 (1986-11).
- KAWA86b 河合, 大村, "映像の言語的表現とコンピュータグラフィックス",
電気関係学会関西支部連合大会, S12-8 (1986-11).
- KAWA88a 河合, 藤田, 大村, "2重節点をもつスプライン基底の一構成法",
信学論(D), J71-D, 6, pp.1149-1150 (1988-6).
- KAWA88b 河合, 山下, 大野, 吉村, 西村, 下條, 宮原, 大村, "並列画像生成
システムLINKS-2のアーキテクチャ", 情処学論, 29, 8,
pp.729-740 (1988-8).
- KAWG83 Kawaguchi, Y., "Growth:Mysterious Galaxy", ACM SIGGRAPH'83,
Film & Video Show, (1983).

- KAWG84 Kawaguchi, Y., "Growth II: Morphogenesis", ACM SIGGRAPH'84, Film & Video Show, (1984).
- KAWG85 Kawaguchi, Y., "Growth III: Origin", ACM SIGGRAPH'85, Film & Video Show, (1985), NICOGRAPH'85, CG Film Show (1985).
- KAWT83 河田, 大野, 吉村, 出口, 西村, 白川, 大村, "コンピュータグラフィックスシステム LINKS-1", 信学技報, IE82-79, pp.39-46 (1983-1-24).
- KAY79 Kay, D. and Greenberg, D., "Transparency for Computer Synthesized Images", Proc. ACM SIGGRAPH'79, 13, 2, pp.158-164 (1979-8).
- KERN78 Kernighan, B.W. and Ritchie, D.M., "The C Programming Language", Englewood Cliffs, New Jersey, Prentice-Hall (1978). 邦訳, "プログラミング言語C - UNIX流プログラム書法と作成", 石田晴久訳, 共立出版(1981).
- KUB080 Kubo, M., Taguchi, Y., Agusa, K. and Ohno, Y., "Multi-Microprocessor System for Three-Dimensional Color Graphics", Proc. IFIP 80, pp.145-150 (1980).
- KUB083 久保, 戸島, 星野, 大野, "マルチマイクロプロセッサを用いたカラーグラフィックスシステム", 情処学論, 24, 4, pp.488-495 (1983-7).
- KUB084 久保, 戸島, 亀井, 阿草, 大野, "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", 情処学論, 25, 1, pp.10-18 (1984-1).
- KUBT87 窪田, 西村, 小林, 中村, 重井, "光線追跡法のための空間分割型並列処理の負荷分散法", 昭62信学総全大, 6, pp.318 (1987-3).
- KURI87 栗山, 河合, 大村, "カーディナルスプラインを用いた自由曲面のモデリング手法", 情処学グラフィックスとCAD研資, 25-3 (1987-2).
- MATS83 松本, 村上, "Octree データ構造を用いた Ray-Tracing法", 情処学

第27回全国大会, pp.1535-1536 (1983).

- MIYA85 Miyagaki, K., "Chasing the Rainbow", ACM SIGGRAPH'85, Film & Video Show (1985), NICOGRAPH'85, CG Film Show (1985).
- MURA86 村上, 佐藤, 広田, "セルラアレイプロセッサCAPによるレイトレーシング", 情処学グラフィックスとCAD研資, 22-2 (1986-7).
- NAKA82 中山, 平井, 大野, 出口, 西村, 江木, 河田, 白川, 大村, "画像生成用マルチマイクロコンピュータシステム", 情処学マイコン研資, 24-2 (1982-11-2).
- NAKG85 Nakagawa, Y., "MAIZ", NICOGRAPH'85, CG Film Show (1985).
- NAKS85 中瀬, 日高, 西村, 宮崎, 野口, 鷺島, "高速浮動小数点演算機能を持つユニットコンピュータ・MCのアーキテクチャ", 情処学計算機アーキテクチャ研資, 58-6, (1985-6-21).
- NEWE72 Newell, M. E., Newell, R. G. and Sancha, T. L., "A Solution to the Hidden Surface Problem", Proc. ACM Ann. Conf., pp.443-450 (1972).
- NEWM79 Newman, W. M. and Sproull, R. F., "Principles of Interactive Computer Graphics", McGraw-Hill, New York, 2nd ed., pp.309-331, 367-410 (1979).
- NIIM84 Niimi, H., Imai, Y., Murakami, M., Tomita, S. and Hagiwara, H., "A Parallel Processor System for Three-Dimensional Color Graphics", Proc. ACM SIGGRAPH'84, 18, 3, pp.67-76 (1984).
- NIIM85 新實治男, "グラフィックス専用マシンの開発動向", 情処学計算機アーキテクチャ研資, 60-2 (1985-12-6).
- NISD84 西田, 辰己, 出口, 西村, 河田, 白川, 大村, "視線探索法に基づく三次元色彩陰影画像生成プログラムの解析", 昭59信学総全大, 6, pp.313-314 (1984-3).
- NISH82 西村, 出口, 大野, 河田, 白川, 大村, 尾崎, "LINKS-1: コンピュータグラフィックスシステム", 情処学マイコン研資, 24-1

(1982-11-2).

- NISH83a 西村, 吉村, 出口, 辰己, 河合, 河田, 白川, 大村, 尾崎, "マルチマイクロコンピュータシステムにおける画像生成のための画面分割方式", 信学技報, CAS82-144, pp.31-36 (1983-2-9).
- NISH83b 西村, 大野, 中山, 江木, 河田, 白川, 大村, "動的変構造マルチマイクロコンピュータシステムと画像生成処理への応用", 信学技報, CAS82-145, pp.37-42 (1983-2-9).
- NISH83c Nishimura, H., Ohno, H., Kawata, T., Shirakawa, I. and Omura, K., "LINKS-1: A Parallel Pipelined Multi-computer System for Image Creation", Proc. 10th Ann. Int. Symp. Comput. Archi., pp.387-394 (1983).
- NISH84 Nishimura, H., Deguchi, H., Kawai, T., Yamashita, S., Kawata, T., Shirakawa, I. and Omura, K., "A Multicomputer System LINKS: Its Architecture and Graphics Applications", Proc. IECON'84, pp.270-274 (1984-10).
- NISH85a 西村, 平井, 河合, 河田, 白川, 大村, "分布関数による物体モデリングと画像生成の一手法", 信学論(D), J68-D, 4, pp.718-725 (1985-4).
- NISH85b 西村, 出口, 辰己, 河田, 白川, 大村, "コンピュータグラフィックスシステムLINKS-1における並列処理の性能評価", 信学論(D), J68-D, 4, pp.733-740 (1985-4).
- NISH85c Nishimura, H., Deguchi, H., Kawai, T., Yamashita, S., Katawa, T., Shirakawa, I. and Omura, K., "A Parallel Processing Scheme for 3D Image Generation and its Applications", Proc. China 1985 ICCAS, pp.280-283 (1985).
- OHTA85 大谷, 新實, 富田, 萩原, "実時間3次元図形表示システムEXPERTSの開発", 信学技報, CAS84-197, pp.53-60 (1985-2-1).
- OKAM81 岡本充由, "3次元画像処理用マルチコンピュータシステムに関する研究", 大阪大学大学院工学研究科電子工学専攻特別研究報告

- (1981-3).
- OMUR86 大村, 河合, "グラフィックス専用マシンLINKS-1, 2",
第17回画像工学コンファレンス, 12-1 (1986-12).
- PATS81 Patterson, D.A. and Sequin, C.H., "RISC I: A Reduced
Instruction Set VLSI Computer", IEEE Computer Architecture
Conf., pp.443-457 (1981).
- PATT85 Patton, C.P., "Microprocessors: Architecture and Applica-
tions", IEEE Comput. Mag., 18, 6, pp.29-40 (1985-7).
- RUBI80 Rubin, S.M. and Whitted, T., "A Three-Dimensional Repre-
sentation for Fast Rendering of Complex Scenes", Proc. ACM
SIGGRAPH'80, 14, 3, pp.110-116 (1980-7).
- RUSS78 Russell, R.M., "Cray-1 Computer System", Comm. ACM, 21, 1
pp.26-31 (1978).
- SAT085a 佐藤, 池坂, 石畑, 柿本, 井上, 石井, "セルラアレイプロセッサC
APのアーキテクチャ", 信学技報, CAS84-200, pp.73-80
(1985-2-1).
- SAT085b 佐藤, 池坂, 石畑, 柿本, 井上, 石井, "セルラアレイプロセッサC
APの応用", 信学技報, CAS84-201, pp.81-88 (1985-2-1).
- SAT085c Sato, H., Ishii, M., Sato, K., Ikesaka, M., Ishihata,
H., Kakimoto, M., Hirota, K. and Inoue, K., "Fast Image
Generation of Constructive Solid Geometry Using A Cellular
Array Processor", Proc. ACM SIGGRAPH'85, 19, 3, pp.95-102
(1985-7).
- SCHA81 Schachter, B.J., "Computer Image Generation for Flight
Simulation", IEEE Comput. Gr. & Appl., pp.29-68 (1981-9).
- SHIM85 沢木, 河合, 那須, 河田, 白川, 大村, "コンピュータグラフィック
ス用画像データベースシステム", 情報学コンピュータビジョン研資,
38-3 (1985-9-25).
- STNZ77 Stenzel, W.J., Kubitz, W.J. and Garcia, G.H.,

- "A Compact High-Speed Parallel Multiplication Scheme", IEEE Trans. Comput., C-26, 10, pp.948-957 (1977-10).
- SUTH63 Sutherland, I. E., "SKETCHPAD: A Man-Machine Graphical Communication System", AFIPS SJCC, pp.329 (1963).
- SUTH74 Sutherland, I. E., Sproull, R. F. and Schumacker, R. A., "A Characterization of Ten Hidden Surface algorithms", ACM Computing Surveys, 6, 1, pp.1-55 (1974-3).
- TAKA87 Takahashi, T., Naruse, T. and Yoshida, M., "Architecture and Performance Evaluation of the Dedicated Graphics Computer: SIGHT", IEEE MONTECH'87(Compint'87), (1987).
- TAMA87 玉邑, 秋本, 三ツ矢, "高速画像生成装置-MAG I C-の動作と性能", 情処学グラフィックスとCAD研資, 27-2, (1987-7).
- TAMA88 玉邑, 秋本, "高速画像生成装置-MAG I C-のCG処理環境と応用", 情処学グラフィックスとCAD研資, 35-7, (1988-9).
- WARN69 Warnock, J., "A Hidden Surface Algorithm for Computer Generated Half-Tone Pictures", Univ. Utah Comput. Sci. Dept., TR 4-5 (1969).
- WHIT80 Whitted, T., "An Improved Illumination Model for Shaded Display", Comm. ACM, 23, 6, pp.343-349 (1980-6).
- WILK69 Wilkes, M. V., "The Growth of Interest in Microprogramming", ACM Computing Surveys, 1, 3, pp.139-145 (1969-9).
- WILL78 Williams, L., "Casting Curved Shadow on Curved Surface", Proc. ACM SIGGRAPH'78, 12, 3, pp.270-274 (1978-8).
- WYLI67 Wylie, C., Romney, G., Evans, D. C. and Erdahl, A. C., "Half-tone Perspective Drawings by Computer", AFIPS FJCC, 27, pp.49-58 (1967).
- YAMA83 山下, 吉村, 西田, 出口, 西村, 河田, 白川, 大村, "コンピュータグラフィックスシステムLINKS-1の現状", 情処学グラフィックスとCAD研資, 10-3 (1983-7-6).

- YAMG82 山口富士夫, "コンピュータディスプレイによる形状処理工学[I],
[II]", 日刊工業新聞社 (1982).
- YAMN84 山名, 西村, 河合, 河田, 白川, 大村, "メタボール技法による自由
曲面の創成と高速画像生成法", 電気関係学会関西支部連合大会,
G14-7 (1984-11).
- YAN85 Yan, J.K., "Advances in Computer-Generated Imagery for
Flight Simulation", IEEE Comput. Gr. & Appl., pp.37-51
(1985-8).
- YOSD85 吉田, 成瀬, 高橋, 内藤, "グラフィックス計算機S I G H Tの基本
構成", 情処学計算機アーキテクチャ研資, 60-4, (1985-12-6).
- YOSD88 吉田, 成瀬, "コンピュータグラフィックス用プロセッサの動向",
情報処理, 29, 10, pp.1109-1114 (1988-10).
- YOSH82 吉村, 辰己, 西村, 河田, 白川, 大村, "L I N K S - 1における画
像生成手法", 情処学マイコン研資, 24-5 (1982-11-2).