

## 大規模図面情報管理における高速処理方式の提案†

鈴木五郎<sup>††</sup> 薄井勝夫<sup>††</sup> 国友佳男<sup>††</sup>  
石賀忠勝<sup>†††</sup> 木之下正美<sup>†††</sup> 野本和之<sup>†††</sup>

対話型の図面編集システムなど図形としての図面情報を取り扱うオンライン設計システムでは、コマンドの応答性を良くするために図面中に配置された各図形の位置情報を管理する必要がある。従来いろいろな管理方式が考えられてきたが、VLSIの論理図面やマスク図面のように非常に大規模な図面の管理には効率の面で問題があることが分かった。そこで大規模な図面へも適用可能な新しい図面情報管理方式であるフィールド・ブロック法を考案した。この管理方式は 1) 大規模な図面を取り扱った場合でも管理情報が極めて少量で済む、2) 管理情報の作成及び変更が極めて容易、などの特徴があり、図面が大規模になればなるほどその効果は大きくなる。フィールド・ブロック法による図面情報管理方式をVLSI用大規模論理図面編集システムで採用してその性能を評価した結果、少量の管理情報で満足な応答時間が得られることが分かり、新管理方式の実用性と有効性を確認することができた。

### 1. はじめに

VLSIの論理設計や実装設計においては、ワーク・ステーションの普及によって設計者自身が直接端末を操作してオンライン設計を行う時代に突入しようとしている。オンライン設計を行う上で最も重要視されるのはコマンドの応答性である。コマンドを投入してから画面上に結果が現れるまでの時間をできるだけ短くすることが設計の流れを中断させずに能率良く作業を進める必要条件となる。オンライン設計を行う対象としては、論理設計では 1) 論理図面編集、実装設計では 1) マスク図面編集、2) 自動コンパクション、3) 設計規則チェックなどがあるが、これらの処理を高速化するためには、図面内のある領域に存在する図形群をできるだけ高速に抽出する必要がある。つまり、論理図面やマスク図面の編集を行う場合、指定したあるウィンドウ内の図形群を移動・回転・ミラー反転・コピー・消去するといった操作を頻繁に行うが、処理時間を決定する一番の要因は図面を構成している全図形群の中からウィンドウ内に存在しているものだけをいかに高速に抽出できるかにある。同様に自動コンパクションや設計規則チェックにおいても、ある図形の付近に存在している図形群を認識する時間が応答性を左右する。いずれの処理においても、図面を構成している各図形が図面中のどこに存在しているかをうまく管理

する方式、つまり図面情報管理方式が全体の処理時間を短くする鍵となっている。従来から各種の図面情報管理方式が考案されてきたが、VLSIに関する莫大な情報(例えば10万トランジスタのオーダ)をオンライン設計しようとするとき良好な応答時間を確保することが困難になることが予測される。そこで我々は大規模図面のオンライン設計に適した図面情報管理方式を提案する。本論文では、最初に従来の図面情報管理方式を、次に筆者らの提案する図面情報管理方式を述べ、最後に性能評価の結果を示す。

### 2. 従来の図面情報管理方式

従来からある図面情報管理方式の代表例として、ソーティング法<sup>1)</sup>と領域分割法<sup>2)</sup>を取り上げ、それぞれの方式の問題点を説明する。図形番号  $g_i$  が割り当てられた6個の矩形(多角形の場合には、外接する矩形を考える)が配置されている図1(a)を例題とする。図1(b)は各図形の実体情報が格納されている実体情報テーブルであり、補助記憶装置と主記憶装置の間で実体情報のスワッピング処理が行われるという前提に立ち、スワッピング処理の単位(ブロック)にグループ化しておく。実体情報テーブルの一つの列が一つのブロックに対応している。入力された各図形の実体情報  $G_i$  は入力順にこのテーブルに格納され、 $G_i$  のヘッダとして矩形の左下、右上座標  $(x_{min}, y_{min})$ ,  $(x_{max}, y_{max})$  が付加される。以下では、図面情報管理方式のいかんにかかわらず実体情報テーブルの構造は変わらないと仮定する。

#### 2.1 ソーティング法

実体情報のヘッダ  $(x_{min}, y_{min})$ ,  $(x_{max}, y_{max})$  を基準

† A Fast Data Management Method for Very Large Scale Schematics by GORO SUZUKI, KATSUO USUI, YOSHIO KUNITOMO (Hitachi Research Laboratory, Hitachi Ltd.), TADAKATSU ISHIGA, MASAMI KINOSHITA and KAZUYUKI NOMOTO (Musashi Works, Hitachi Ltd.).

†† (株)日立製作所日立研究所

††† (株)日立製作所武蔵工場

として図形番号  $g_i$  を図 2 (b) のようにソーティングしておく方式である。この例では図形番号とその実体情報が格納されている実体情報テーブルのブロック番号との対応をとるための対応テーブルを持たせてある。

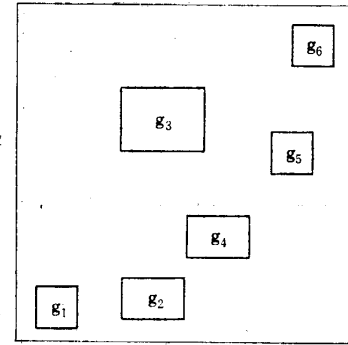
この方式は図面規模が小さい場合には有効な手段となるが、規模が大きくなると次の点で問題を生じる。

(1) 管理情報の作成・変更

図形を追加あるいは既入力の図形を変更すると共に管理情報テーブルをソーティングし直さなければならないが、図形数が多くなるに従ってこの処理に莫大な時間を費やすことになる。例えば図 2 (c) のように図形  $g_3$  と  $g_4$  を移動した場合、管理情報テーブルを図 2 (d) のようにすべて変更する必要がでてくる。

(2) 管理情報量

各図形に関して  $(x_{min}, y_{min})$ ,  $(x_{max}, y_{max})$  の 4 個の座標でソーティングすることから、図面の規模によって管理情報量が比例して増加する。管理情報は高頻度で参照・作成・変更されるものであるから、極力主記憶装置上に常駐させたほうが良



(a) 図面

$b_1$	$h_1$	$G_1$	$h_2$	$G_2$	$h_3$	$G_3$
$b_2$	$h_4$	$G_4$	$h_5$	$G_5$	$h_6$	$G_6$

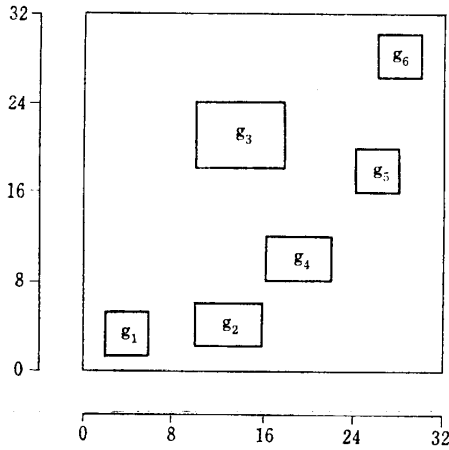
(b) 実体情報テーブル

$g_1 \sim g_6$ : 図形番号  
 $b_1, b_2$ : ブロック番号  
 $G_1 \sim G_6$ : 図形実体情報  
 $h_1 \sim h_6$ : ヘッダ  
 $(x_{min}, y_{min}) (x_{max}, y_{max})$

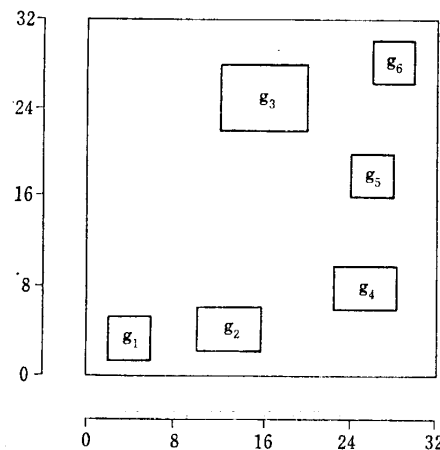
図 1 例題図面

Fig. 1 Example schematic.

い。このため管理情報量が多くなるほど主記憶装置上に常駐できる実体情報の量は少なくなり、ス



(a) 図面



(c) 変更後の図面

対応テーブル

$b_1$	$g_1$	$g_2$	$g_3$
$b_2$	$g_4$	$g_5$	$g_6$

$b_1, b_2$ : ブロック番号  
 $g_1 \sim g_6$ : 図形番号

X座標でのソーティング		Y座標でのソーティング	
:	:	:	:
16	$g_2$	6	$g_2$
16	$g_4$	8	$g_4$
18	$g_3$	12	$g_4$
22	$g_4$	16	$g_5$
24	$g_5$	18	$g_3$
26	$g_6$	20	$g_5$
:	:	:	:

(b) 管理情報テーブル

:	:	:	:
12	$g_3$	6	$g_4$
16	$g_2$	10	$g_4$
20	$g_3$	16	$g_5$
22	$g_4$	20	$g_5$
24	$g_5$	22	$g_3$
26	$g_6$	26	$g_6$
:	:	:	:

□: 変更箇所

(d) 変更された管理情報テーブル

図 2 ソーティング法  
 Fig. 2 Sorting method.

ワッピングを行う確率が高くなる。つまり図面が大規模になるほど、必要な実体情報の抽出に時間がかかることになる。

2.2 領域分割法

図3(a)のように図面をあらかじめ格子状の領域  $R_i$  に分割しておき、各図形がどの領域に存在しているかを管理する方式である。例えば図3(b)のように各領域ごとにそこに存在する図形情報が格納されている実体情報テーブルのブロック番号を管理しておく。この方式もソーティング法と同じように図面の規模が大きくなると次のような問題点が生じる。

(1) 管理情報の作成・変更

例えば図3(c)のように図形  $g_3$  と  $g_4$  を移動した場合、図形が存在しなくなった領域と新たに存在するようになった領域の管理情報を変更しなければならない(図3(d))。

(2) 管理情報量

必要な実体情報の抽出を高速化しようとして領域分割を細かくすると、複数の領域にまたがる図形の個数が増加し管理情報量が莫大になってしまう。そこで前述したのと同じ理由でスワッピングの頻度が高くなり、必要な実体情報の抽出に時間がかかることになる。

3. 提案する図面情報管理方式

従来の図面情報管理方式では、図面の規模が大きくなると、必ずしも十分な応答時間が得られなくなることが分かった。そこで我々は管理情報量をできるだけ少なくし、かつその作成・変更が容易になるフィールド・ブロック法を考案した。

3.1 基本概念

フィールド・ブロック法は、実体情報テーブルのブロックごとにその物理的な広がりを持つことにより管理を行う方式である。ブロックの物理的な広がりには次に定義する4個の座標  $(X_{min}, Y_{min})$ ,  $(X_{max}, Y_{max})$  で表現する。

$$X_{min} = \min \{x_{min}\}$$

$$Y_{min} = \min \{y_{min}\}$$

$$X_{max} = \max \{x_{max}\}$$

$$Y_{max} = \max \{y_{max}\}$$

ここで、 $(x_{min}, y_{min})$ ,  $(x_{max}, y_{max})$  は該当するブロック内に格納されている各図形  $G_i$  のヘッダつまり左下、右上座標である。

このように物理的な広がりを持つブロックのことをフィールド・ブロックと呼ぶことにする。例題図面では図4(a)のように破線で表現した二つのフィールド・ブロックが存在することになる。管理情報としては図4(b)のように各ブロックの物理的な広がり情報  $(X_{min}, Y_{min})$ ,  $(X_{max}, Y_{max})$  だけである。図面中で指定したあるウィンドウ内の図形群をどのような手順で抽出するかを図5を用いて説明する。管理情報テーブルには各ブロック情報が実体情報テーブル上のどこに存在しているか、また実体情報テーブル上にはない場合には補助記憶装置上のどこに存在しているかを表す二つのポイント TBPT と DBPT が追加されている。以下に手順を示す。

(手順1)

ウィンドウの左下、右上座標

$(X_{MIN}, Y_{MIN})$ ,  $(X_{MAX}, Y_{MAX})$

が指定される。

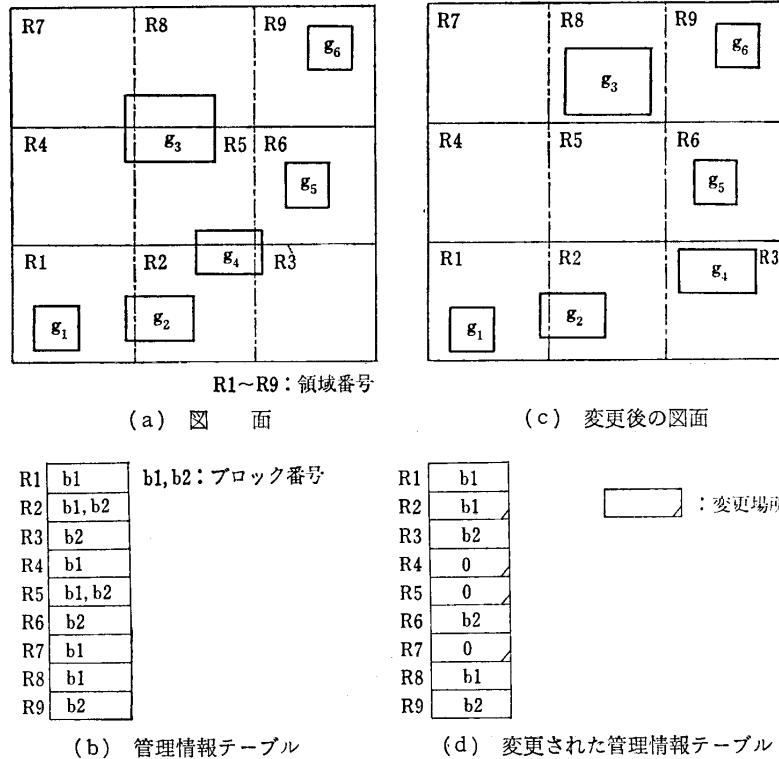
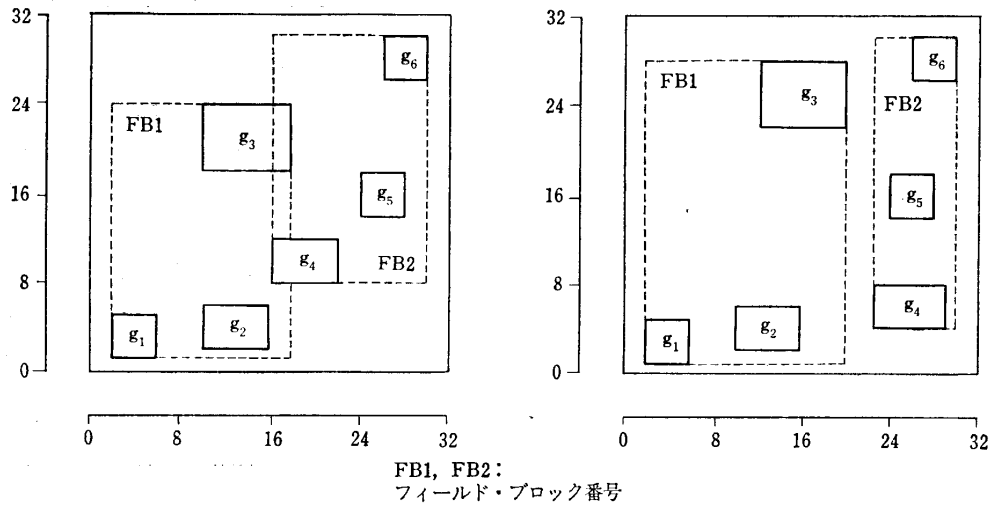


図3 領域分割法  
Fig. 3 Area partition method.



	$(X_{min}, Y_{min})$	$(X_{max}, Y_{max})$
b1	( 1, 1)	(18, 24)
b2	(16, 8)	(30, 30)

b1, b2: ブロック番号

	$(X_{min}, Y_{min})$	$(X_{max}, Y_{max})$
b1	( 1, 1)	(20, 28)
b2	(22, 4)	(30, 30)

◻ : 変更箇所

(b) 管理情報テーブル

(d) 変更された管理情報テーブル

図 4 フィールド・ブロック法  
Fig. 4 Field block method.

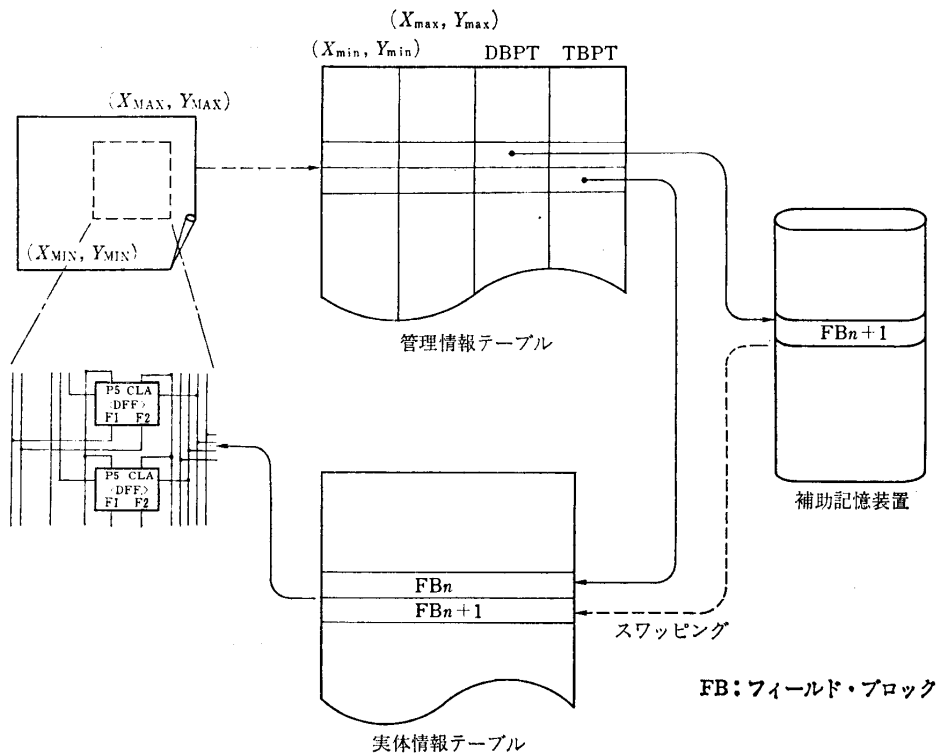


図 5 図形情報抽出手順  
Fig. 5 Data searching process.

## (手順2)

管理情報テーブルに格納されている各フィールド・ブロックの  $(X_{min}, Y_{min})$ ,  $(X_{max}, Y_{max})$  から、指定されたウィンドウにかかっている全フィールド・ブロックの番号を抽出し、ブロック情報が格納されている主記憶装置上または補助記憶装置上の位置を求める。

## (手順3)

(手順2)で求めた一つのフィールド・ブロックに注目し、それが実体情報テーブル上に存在している場合には(手順4)へ。存在していない場合には補助記憶装置上の該当ブロックを実体情報テーブル上に持ち込む。ここで、実体情報テーブル上のどのブロックを無効にするかは例えば LRU 法<sup>3)</sup> (Least Recently Used 法) により決定する。

## (手順4)

フィールド・ブロック内に格納されている各図形に注目し、そのヘッダ情報  $(x_{min}, y_{min})$ ,  $(x_{max}, y_{max})$  からウィンドウにかかる図形群を抽出する。図形が多角形である場合、さらに図形の詳細な形状からウィンドウにかかるか否かを判定する。

## (手順5)

(手順2)で求めたすべてのフィールド・ブロックに関して(手順3)(手順4)を繰り返す。

## 3.2 従来方式との比較

## (1) 管理情報作成・変更

例えば図4(c)のように図形  $g_3$  と  $g_4$  とを移動した場合を考える。管理情報の変更は、図4(d)のように移動した図形が属しているフィールド・ブロックの広がりに変化があるときのみ行う。変更する内容は  $(X_{min}, Y_{min})$ ,  $(X_{max}, Y_{max})$  だけであるから極めて簡単な処理になる。

図形を追加・回転・ミラー反転・消去した場合も管理情報の変更は同じ要領である。

## (2) 管理情報量

図2(b), 図3(b)と図4(b)を比較すれば、フィールド・ブロック法における管理情報が極めて少量であることが分かる。つまり、量的に管理情報が実体情報を圧迫しスワッピングの頻度を高めてしまうような事態は生じない。

以上の比較から、フィールド・ブロック法はVLSIのように非常に莫大な情報をオンライン設計する場合に極めて有効な図面情報管理方式であることが分かる。ただし、この方式にも弱点は存在する。つまり図形の入力順序によっては図6(a)のようにフィールド・ブロックがむやみに大きく広がってしまう場合がある。このような問題に対処するため我々はフィールド・ブロックの自動分割処理を施している。フィールド・ブロックがある大きさ以上になった場合には図6(b)のようにできるだけその広がりを小さくすべく複数のフィールド・ブロックに自動分割するのである。

## 4. プログラム評価

フィールド・ブロック法をプログラム化し、VLSI用大規模論理図面編集システム DASH<sup>4)</sup> (Design Automation System for Hierarchical Schematics) に組み込んでその評価を行った。プログラム言語は FORTRAN 77, 使用計算機は HITAC M200 H (約 8 MIPS) である。

## 4.1 DASH システム

DASH は VLSI 論理のオンライン設計システムであり、論理図面を図形として入力することによってシミュレータや自動配置・配線など各種 CAD プログラムの入力データを自動生成することを目的としてい

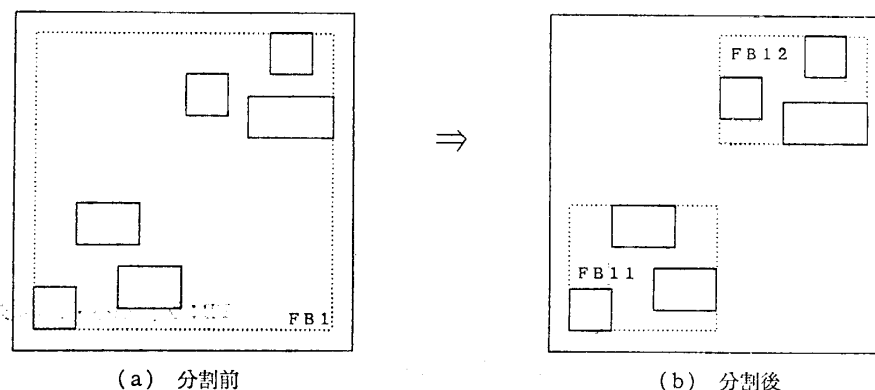


図6 フィールド・ブロックの分割  
Fig. 6 Field block partition.

る。図7にシステム構成を示す。DASHは大型計算機、及びこれと通信回線で接続された図形入出力端末 HITAC G-760<sup>5)</sup> を使用している。図形入力としてタブレットとスタイラスを使い、入力されたコマンドやパラメータの解釈を端末のマイクロ・コントローラで行う。また端末には最大2Mバイトのローカル・メモリーが装備されており、拡大・縮小・スクローリングなどの表示制御が高速に行える。DASHシステムは1) 図形としての図面情報を編集する図面編集プログラム 2) 図面情報から共通ネット情報(特別のCADプログラムに依存しない)を抽出するネット抽出プログラム 3) 共通ネットからシミュレータや自動配置・配線など各種CADプログラムの入力データを編集するCAD入力編集プログラムなどから構成されている。図面情報データ・ベースに格納されている図面情報がフィールド・ブロック法により管理されている。

4.2 性能評価 (1)

次の条件で性能評価を行った。

- (1) 実体情報テーブルの1ブロックの大きさは6144バイトである。
- (2) 図形の実体情報は約200バイトであり、一つのフィールド・ブロックには約30個の図形が存在している。
- (1) 管理情報量の評価

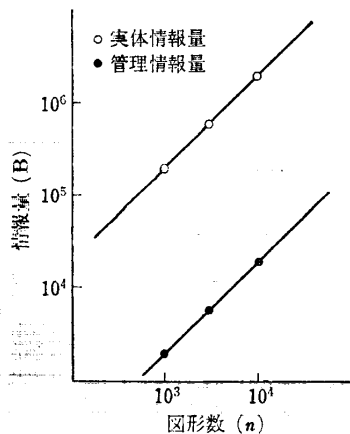


図8 実体情報量と管理情報量  
Fig. 8 Data demands for schematic and management.

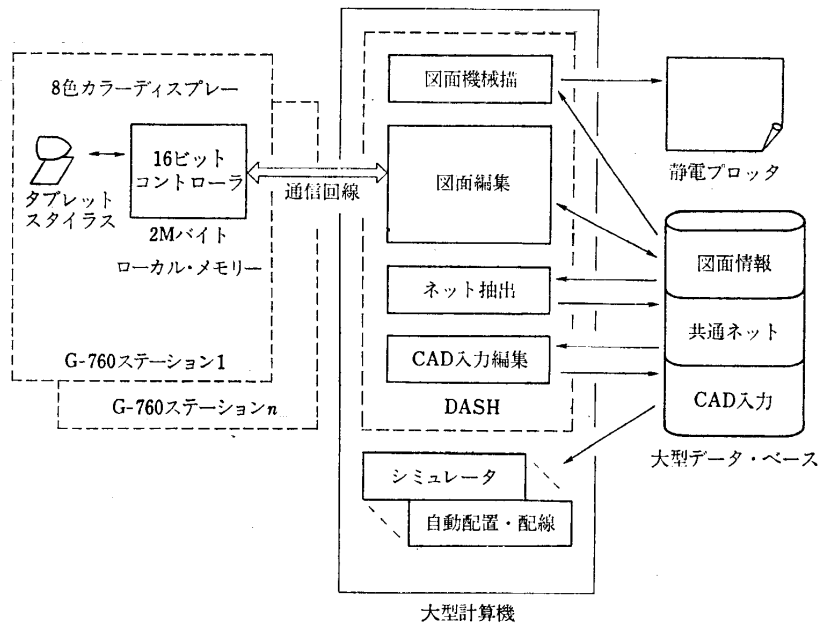


図7 システム構成  
Fig. 7 System configuration.

図8は横軸が図形数を、縦軸が管理情報量と実体情報量を表している。10k個の図形を扱った場合でも実体情報が2Mバイトなのに対して管理情報量は20kバイトであるから、その比率は1%でありほとんど無視できる値である。

(2) ウィンドウ内図形抽出処理時間の評価

図9は図面中に存在する図形の個数と、ウィンドウを指定してその内部に存在する図形を抽出した時のCPU処理時間を表している。ここでは一つの図形のみがかかるようにウィンドウを指定している。処理速度は図形の個数nに対して $O(n^{1/5})$ 程度の変化である。1k個の図形の中からの選択でも0.022秒程度であり極めて高速に処理できることが分かる。

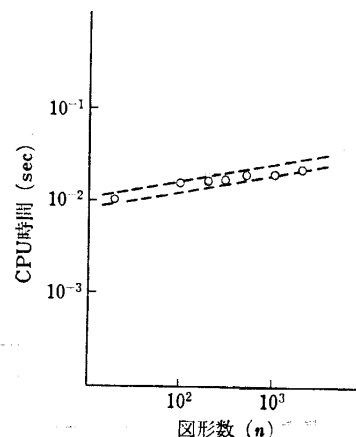


図9 図形抽出処理時間  
Fig. 9 CPU time for selecting elements.

### (3) 領域分割法との比較

0.6k 個の図形を配置した図面を用いて、領域分割法とフィールド・ブロック法とで管理情報量とウィンドウ内図形抽出処理時間がどのように異なるかを実験した。表1がその結果を表している。フィールド・ブロック法は領域分割法と比べてウィンドウ内図形抽出処理時間において約18%、管理情報量において約13%で済んでいる。

### 4.3 性能評価 (2)

DASH システムの持つ図面編集用コマンドの中で図面情報管理方式に深く関連するものに注目し、その応答時間を測定した。コマンドの応答時間はコマンド投入時点からグラフィック・ディスプレイ上に最初の反応が現れるまでの時間とした。測定を行ったときの計算機環境を下記に示す。

(1) アクティブ端末 …平均 120 台

(2) 大型計算機とG-760との接続 …9600bps

実験には 0.7k 素子 (ゲート数+端子数) と図10に示した 1.6k 素子からなる図面を使用した。表2に示したコマンドの応答時間は各コマンドを 10 回動かした時の平均である。

ウィンドウ選択コマンドに関しては、10個余りの図

表 1 領域分割法との比較

Table1 Comparison with the area partition method.

	管理情報量 (kB)	処理時間 (秒)
領域分割法	9.75	0.11
フィールド・ブロック法	1.28	0.02

形群を選択し、選択されたという表示が現れるまでの時間である。結果は各図面とも 3 秒以下の応答時間である。

素子配置・移動・回転及び消去コマンドに関してはあらかじめ 1 個の素子を選択しておき、それぞれの処理を行った結果がディスプレイ上に表示されるまでの時間である。ここでは、主に管理情報作成・変更に関する処理時間が反映されている。実験結果は、各図面ともに 1 秒～2 秒程度の応答時間である。

これらの結果から図面情報管理方式に深くかかわるコマンドの応答時間は大規模な図面になっても良好であることが分かる。

### 5. む す び

新しい図面情報管理方式であるフィールド・ブロッ

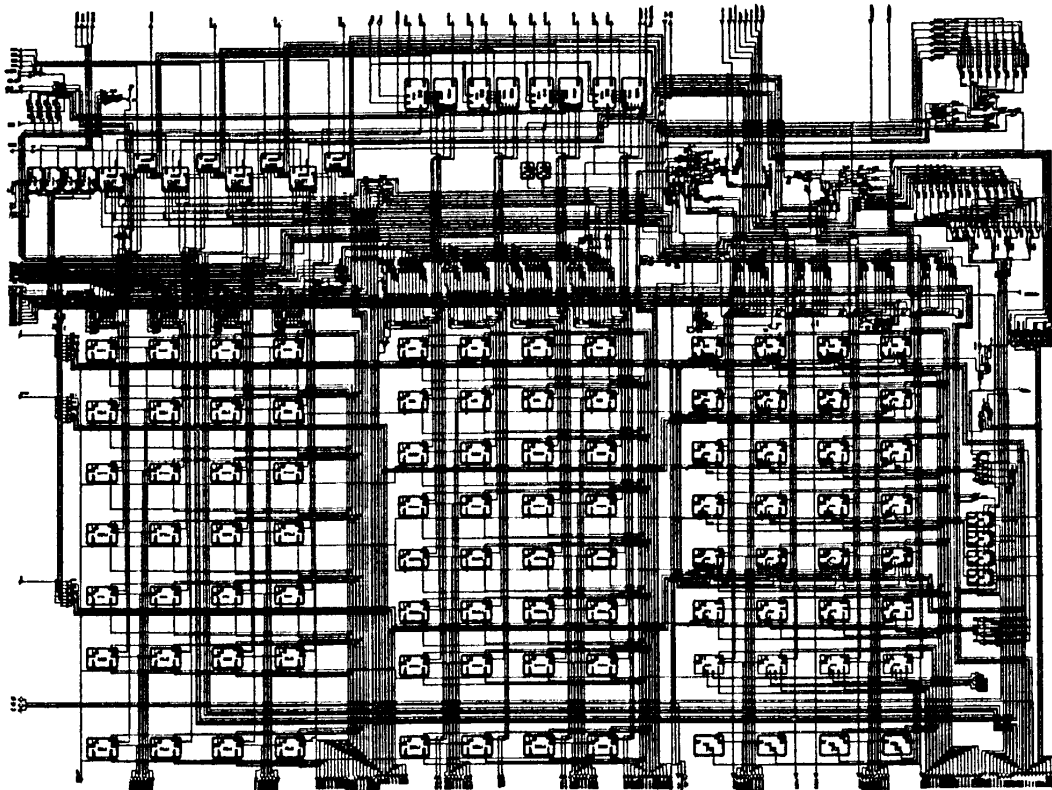


図 10 論理図面

Fig. 10 Logical schematic.

表 2 図面編集用コマンドの応答時間 (単位: 秒)  
Table 2 Schematic edit command response time.

	ウィンドウ選択	素子配置	移動	回転	消去
1600 素子	2.8	1.1	1.7	1.5	1.2
700 素子	2.1	0.8	1.2	1.3	1.1

ク法を開発した。この方式は大規模な図形を扱った場合でも管理情報は極めて少量であり、かつ管理情報の作成及び変更が容易に行える特徴を持つ。

フィールド・ブロック法をプログラム化し、VLSI 用大規模論理図面編集システムに組み込んで評価を行った。その結果、管理情報は実体情報の1%で済むこと、及び1.6k素子から構成されている大規模な図面の編集においても代表的なコマンドの応答時間が1秒～3秒と良好な結果を得ることを確認した。

ここで提案した新しい図面情報管理方式はVLSIのCADシステムだけに限らず、地図の地理コード・システムであるとか機械系・建築系の図面編集システムなど図形としての図面情報を取り扱う幅広いCAD分野に適用することができ、その応用範囲は極めて広い。

謝辞 本研究を行うに当たり、ご指導・ご助言をいただいた日立製作所谷中雅雄部長、福田秀樹部長はじめご討論いただいた関係諸氏に深く感謝します。

### 参 考 文 献

- 1) Baird, H. S.: Fast Algorithm for LSI Artwork Analysis, 13th DAC, pp. 303-311 (June 1976).
- 2) Wilmore, J. A.: The Design of an Efficient Data Base to Support an Interactive LSI Layout System, 16th DAC, pp. 445-451 (June 1979).
- 3) 益田: ペーシング・マシンにおけるスワッピング・アルゴリズム, 情報処理, Vol. 13, No. 2, pp. 81-88 (1972).
- 4) 鈴木: VLSI 用階層論理設計システム (DASH), 信学技法, SSD 84-79, pp. 43-49 (1984).
- 5) Nio, M.: Distributed Processing Terminal for CAD/CAM, Compsac, pp. 547-552 (1979).

(昭和60年1月14日受付)

(昭和60年12月19日採録)



鈴木 五郎

昭和26年生。昭和50年慶応義塾大学工学部電気工学科卒業。同年(株)日立製作所入社。以来マン・マシン・インターラクティブ・システムを中心としたVLSI・CADの研究

究・開発に従事。現在同社日立研究所第3部研究員。IEEE, 電子通信学会各会員。



薄井 勝夫 (正会員)

昭和19年生。昭和37年日立工業専修学校機械科修了。同年(株)日立製作所入社。計算機制御システム、ソフトウェア工学、CADシステムの研究開発を経て、昭和60年より日立エンジニアリング(株)に勤務。現在、VLSIのCADシステムの開発に従事。

国友 佳男 (第27巻第1号参照)



石賀 忠勝 (正会員)

昭和19年生。昭和41年東京大学教養学部基礎科学科卒業。昭和43年同理学系大学院物理修士課程修了。同年4月(株)日立製作所に入社。中央研究所に配属される。ネットワーク、回路シミュレーション、レイアウト等半導体向けCAD開発に従事。その後、電子線描画装置の制御プログラムの開発を担当。昭和57年武蔵工場に移り、対話型論理図/回路図編集システム、高速回路シミュレータ、素子モデリングなどを担当。昭和61年本社生産技術部ソフト技術推進センタに勤務。



木之下正美

昭和31年生。昭和54年電気通信大学電気通信学部情報数理工学科卒業。同年(株)日立製作所入社。現在同社半導体事業部武蔵工場DA開発部でLSI用CAD/DAの開発に従事。



野本 和之

昭和34年生。昭和55年東京工業高等専門学校電子工学科卒業。同年(株)日立製作所入社。現在、同社半導体事業部武蔵工場DA開発部でLSI用CAD/DAの開発に従事。