

マスクパターンの図形演算手法

築添 明 三浦 地平 小澤 時典
((株) 日立製作所 中央研究所)

1. 緒 言

LSIの大規模化に伴ない開発期間の長くなる原因の一つは、設計不良によるマスク再製作、LSI再試作である。これを減らすために各種CADシステムが開発されてきた。

LSIの設計情報は最終的にはマスクパターンデータに集約される。マスクパターンデータを対象としたCADシステムには図形演算処理が必須であり、そのシステムの機能、性能は図形演算処理の機能、性能に左右される。図形演算とは、マスクパターン上の図形間の重なり関係を認識し、重なった部分(AND)を取り出すとか、輪郭どり(OR)をする操作である。図形演算処理の技術的課題は大量のマスクパターンデータを高速に処理する手法の開発であり、国内外で多数の図形演算手法が提案されてきた。

本論文では、従来の図形演算手法の整理と比較検討、および、高速化手法の提案を行なう。提案する手法の処理時間と入力データ規模の関係についても検討を加える。

2. 図形演算処理

2.1 対象図形

図形演算の対象とする図形は図2.1に示すものである。マスクパターンの図形としては、直交図形(すべての辺がxまたはy座標軸に平行である図形)と斜め図形(少なくとも一辺が両座標軸に平行でない図形)とで十分であろう。

対象図形の特殊例を図2.2に示す。これらはデータの入力過程で現実には生じてくるもので、このような特殊図形を含むデータも取扱えなければならぬと考える。

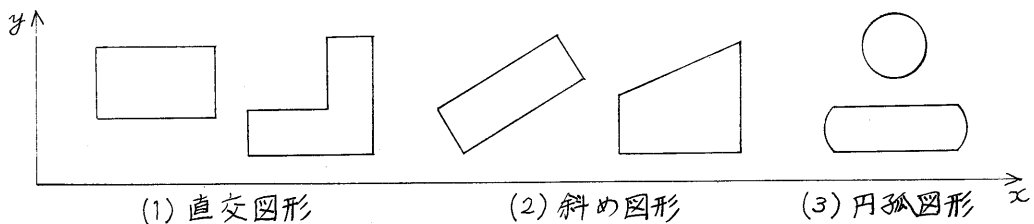
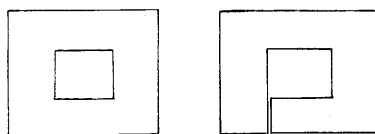
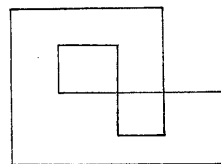


図2.1 対象図形

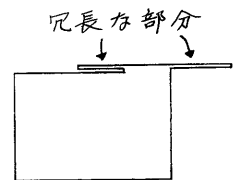
(2多角形で表現) (一筆書き)



(1) 白抜き図形



(2) 自己交差型図形



(3) 冗長な表現の図形

図2.2 対象図形の特殊例

2.2 演算機能

図形演算の機能は、1) 論理演算、2) 位相演算、3) 分解演算、4) 幾何計算、に大別される。これらの演算機能は図2.3～図2.6に示すものである。

O R	
AND	
SUB	
EOR	
NOT	
NOR	
NAND	

図2.3 論理演算

RECT	
台形	

図2.5 分解演算

CONTAIN	
MEET	
DISJOINT	
TOUCH	

図2.4 位相演算

BROADEN	
図形間の距離	
図形の幅	
図形の面積	
図形の周辺長	

図2.6 幾何計算

3. 図形演算手法

図形演算手法の従来技術を30文献[1]~[30]について調査分類し、表3.1にまとめた。本章では最初に、図形演算手法と密接な関係にあるデータの表現法について述べる。つぎに、図形演算手法を1)演算対象の選択法、2)選択後の論理演算手法(図形間の重なり関係の認識手法)、の2面から分類整理する。最後に、従来手法を比較検討してまとめる。

3.1 図形演算データの表現法

図形演算データの表現法は、1)多角形表現(POLYGON)、2)方形表現(RECT)、3)ベクトル表現(VECTOR)、の3通りがある。計算機内部での記憶形式も含めて図3.1に示す。以後に述べる図形演算手法でのデータ表現法は、1)~3)のいずれかを使用している。

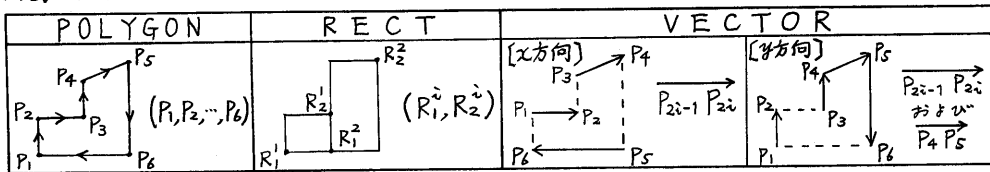


図3.1 図形演算データの表現法

3.2 演算対象の選択法

大規模なマスクパターンデータに対しては、図形演算を施すべき対象図形を取り出す操作が処理時間の多くを占める可能性があり、その選択法が問題となる。従来方法は、1)単純比較法[1]、2)タッチング法[8]、3)スリット法[10]、の3通りである。これらの原理的処理法と処理時間 T とデータ数 N との関係をまとめたものが図3.2である。ビットマップ法[12]は、スリット法の考え方を y 方向にも適用し分割された格子に注目することから、スリット法に含めた。

手法名	原理的処理法	説 明	処理時間 評価式
単純比較法	入力順に注目し、注目データ以外のまだ注目していないデータを対象として選択する。	 n : 入力順 i に注目している時は、 $i+1, i+2, \dots, 5$ のデータを対象として選択する。	$T \propto \frac{N^2 - N}{2}$
タッチング法	注目データの x 方向幅(右図斜線部)に入っているデータを対象として選択する。	 n : 注目順 1に注目している時は、2, 3, 4のデータを対象として選択し、5は選択しない。	$T \propto N^{1.5}$ (斜線部) $\propto \sqrt{N}$ (データ数)
スリット法	隣接する x 座標間の領域(右図斜線部でスリットと呼ぶ)に注目し、スリット内の x 方向ベクトルを対象として選択する。	 太線の x 方向ベクトルを対象として選択する。 (ビットマップ法)	$T \propto N \log N$

図3.2 演算対象の選択法

表 3.1 図形演算手法の分類

(注1) ○：可
 ×：不可
 (手段、データ形式、
 からの指定を含む)
 虫欄：不明

(注2)「分類」
 D：設計ルールチェック CADの文献
 L：論理接続
 C：回路特性
 E：電子線描画装通用
 A：図形演算手法の文献

(注3)「対象図形」白抜き図形については、記述している文献がほとんどなかったため、記述がなければ備考欄に示した。

(注4)「幾何計算その他」A：面積、L：周長
 (注5)「演算対象選択法」③：ビットマップ法

参考文献 No.	発表 年	発表者	プログラム名 or コマンド名 or アルゴリズム名	分類	対象図形			演算機能										図形演算手法				備考																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																									
					直交 交差 交差型	円 長 表現	自己 交差 表現	論理演算	位相演算	幾何計算	データ表現	演算対象 選択法	論理演算 手法	処理時間 評価式																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
															AND	OR	ASUOの他	CONTAIN	MEET	DISJOINT	TOUCH		RECT	台形	ROADEN	距離、他 幅	POLYGON	RECT	VECTOR エカ方向 スカ方向	単純比較法	ランダム 選択法	交点追跡法	ベクトルの向き																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																														
1	72	Redac Software		D	○			○	○	○	○	○	○	○	○	○	○	○	○	○																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																											

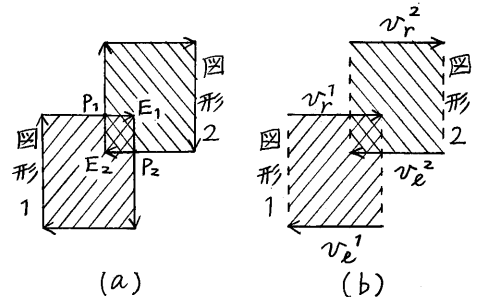
参考文献 No.	発表 年	発表者	プログラム名 or コマンド名 or アルゴリズム名	分類	対象図形		演算機能			図形演算手法			備考	
					直交型 斜交型 円弧 交差表現	円弧 交差表現	論理演算	位置演算	分解演算	幾何計算	データ表現	演算対象 選択法		演算手法
13	77	RCA	Vertex scanning method	A	○	○	○	○	○	○	○	○	TOC N1.5 (N:頂点数) NlogN O(N ^{1.5}) (N:頂点数)	理論的プログラムの 校対評価結果
14	77	東芝	LCS	D	○	×	○	○	○	○	○	○	TOC N1.2 (N:素子数)	
15	77	日立	IVS	L	○	○	○	○	○	○	○	○		
16	77	日立	IVS	L	○	○	○	○	○	○	○	○		
17	77	日立	IVS	L	○	○	○	○	○	○	○	○		
18	78	武蔵野通研 日電	MAICAP	C	○	○	○	○	○	○	○	○		
19	78	超LSI研	AMDES	E	○	○	○	○	○	○	○	○		
20	78	Bell-Northern Research	PLOTCHECK	D	○	×	○	○	○	○	○	○	TOC N1.16 (N:ピクセル内ベクトル総和)	
21	78	ECL Division of E-Systems		A	○	×	○	○	○	○	○	○	*45°のみ	
22	78	NCA	DRC	D	○	×	○	○	○	○	○	○	TOC A1.25 (A:素子数)	
23	79	松下	PANAMAP-R	C	○	×	○	○	○	○	○	○		
24	79	日電		A	○	×	○	○	○	○	○	○	白抜き図形は 一筆書きのみ可	
25	79	日電	PALMS	L	○	○	○	○	○	○	○	○	*方形, 三角形, 線分の3種	
26	79	松下		C	○	○	○	○	○	○	○	○		
27	79	IBM	USC	D	○	○	○	○	○	○	○	○	TOC N (N:図形数)	
28	79	IBM	BDR-T	D	○	○	○	○	○	○	○	○	USCコマンド 使用	
29	79	Department of Defence		D	○	○	○	○	○	○	○	○		
30	79	武蔵野通研	PAS-1	C	○	×	○	○	○	○	○	○	TOC N (N:素子数) *45°のみ	

3.3 論理演算手法

演算対象として選択された複数図形またはベクトルから図形間の重なり関係を認識する手法は、基本的な考え方としては1) 交点追跡法[4]、2) ベクトルの向きによる重なり判定法、の2つがある。実現方法としては、データの表現法や演算対象の選択法と関連して種々の変化がある。

1) 交点追跡法

図3.3(a)の2図形に対して、2図形間の交点 P_1 、 P_2 を求める。図形1の領域内(斜線部)に入り込む図形2の辺 $\overrightarrow{P_2E_2}$ 、図形1の領域内から出ようとする図形2の辺 $\overrightarrow{E_2P_1}$ 、図形2の領域内に入り込む図形1の辺 $\overrightarrow{P_1E_1}$ 、図形2の領域内から出ようとする図形1の辺 $\overrightarrow{E_1P_2}$ により重なり部分が構成されると判定する。



2) ベクトルの向きによる重なり判定法

図3.3 論理演算手法の説明図

図3.3(b)の2図形に対して、図形1の領域は、左向きベクトル v_e^1 とそれに対応する右向きベクトル v_r^1 との間にありと認識する。図形2の左向きベクトル v_e^2 は図形1の領域内にあり、右向きベクトル v_r^2 は図形1の領域外にあるので、 v_e^2 と v_r^1 との間に重なり部分があると判定する。

3.4 従来手法のまとめ

- (1) 対象図形： 77年以後、ほとんどが斜め図形を取扱える。しかし、自己交差型および冗長な表現の図形を取扱えると明記しているものはない。
- (2) 演算機能： 各CADの目的に添った機能を持っているが、汎用的に使える各種機能をコマンド群として持つものがいくつかある[7],[10],[14],[27]。
- (3) データの表現法： POLYGON、RECT表現のものが多く、各CADのデータ表現法や図形演算手法と密接に関連しているものと思われる。
- (4) 演算対象の選択法： 処理時間についてみると、スリット法が最も有利と考えられる。ビットマップ法は、より細分化しているため重なり判定処理が増加する欠点がある。
- (5) 論理演算手法： 処理時間についてみると、交点追跡法の方が不利と考えられる。処理単位が2図形に限定されること、交点計算が必要なことによる。処理単位が2図形に限定されると、 n 個の図形が処理対象の場合($n-1$)回の論理演算が必要である。2図形間の交点計算すべき辺の対の選択法は、最も不利な単純比較法になる。

4. 高速化手法

大規模データを高速処理する手法を立案し、ベクトルスライス法と名づけた。本手法は、演算対象選択法としてスリット法、論理演算手法としてベクトルの向きによる重なり判定法を採用している。

4.1 対象図形

マスクパターンで使用されるすべての多角形データを取扱える。直交図形、任意角度の斜め図形、白抜き図形、自己交差型図形、冗長な表現の図形である。

4.2 適用できる図形演算機能

ベクトルスライス法はすべての図形演算に適用できる。

4.3 図形演算データの表現法

図形演算データはス方向ベクトル、すなわち、水平ベクトルと斜めベクトルとで表現する。(図4.1(1)参照)

4.4 スリットの定義

図形演算入力ベクトルの左右の端点のx座標とベクトル間の交点のx座標との集合を X_s とすると、 X_s 内の隣接するx座標間の領域をスリットと呼ぶ。

4.5 アルゴリズム

論理演算の場合のアルゴリズムを示す。演算結果の出力図形もス方向ベクトルで表現するとする。

ステップ1: スリット境界x座標の集合 X_s を求める。

ステップ2: 入力ベクトルをスリットにより分割する。

ステップ3: 注目スリット S_i を決める。

ステップ31: スリット S_i 内にある分割されたベクトルの集合を求める。

ステップ32: 演算対象ベクトルを決める。

ステップ321: ベクトルの向きから上下の領域の重なり数を計算し、計算値を比較して演算対象ベクトルが出力図形を構成するベクトルであるか否かを判定する。(図4.1(2)参照)

ステップ322: まだ対象にしていないベクトルがあれば、ステップ32にもどる。

ステップ33: まだ注目していないスリットがあれば、ステップ3にもどる。

ステップ4: 出力図形を構成するベクトルをつないで出力ベクトルを求める。

(図4.1(3)参照)

位相演算、分解演算、幾何計算の場合のアルゴリズムも同様である。

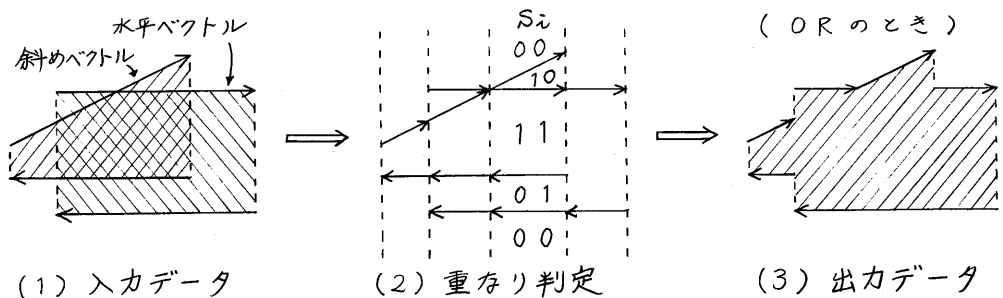


図4.1 ベクトルスライス法

4.6 処理時間の理論的検討

ベクトルスライス法の処理時間 T と入力ベクトル数 N_v との関係を理論的に検討する。

[記号の定義]

N_p ----- 入力図形の頂点数

D_p ----- 頂点の密度

l_{cx}, l_{cy} ----- チップのx方向辺長, y方向辺長

$\overline{l_v}$ ----- ベクトル長平均値

$\overline{w_s}$ - - - - - スリット幅平均値

N_s - - - - - スリット数

$\overline{N_{vs}}$ - - - - - スリット内ベクトル数平均値

前節のアルゴリズムにおいてステップ32~322は最も繰返し回数が多く、他のステップに比して処理時間が十分大きいと考えられる。したがって、

$$T \propto N_s \times \overline{N_{vs}} \quad (\text{式A})$$

と近似できる。最初につきの仮定をおく。

仮定1: 頂点はチップ全体に密度 D_p で分布している。

仮定2: チップは正方形である。 $l_{cx} = l_{cy} = l_c$ とする。

1) 斜めベクトルがない場合

頂点の degree は斜めベクトルがないからすべて1であり、

$$N_p = 2 \times N_v \quad (\text{式1})$$

である。仮定2より

$$D_p = N_p / l_c^2$$

であり、式1を代入して

$$l_c = \{2 \times N_v / D_p\}^{\frac{1}{2}} \quad (\text{式2})$$

となる。スリット数は、仮定1、式2より

$$\begin{aligned} N_s &= \{N_p / (\overline{w_s} \times l_c \times D_p)\} \\ &= \{\sqrt{2} / (\overline{w_s} \times \sqrt{D_p})\} \times \sqrt{N_v} \end{aligned} \quad (\text{式3})$$

である。スリット内ベクトル数平均値は、仮定1、式2より

$$\begin{aligned} \overline{N_{vs}} &= \overline{l_v} \times l_c \times D_p / 2 \\ &= \{\overline{l_v} \times \sqrt{D_p} / \sqrt{2}\} \times \sqrt{N_v} \end{aligned} \quad (\text{式4})$$

である。式3、式4より

$$N_s \times \overline{N_{vs}} = \{\overline{l_v} / \overline{w_s}\} \times N_v \quad (\text{式5})$$

となる。 $\overline{l_v} / \overline{w_s}$ は、ベクトルのスリットにより分割される数の平均値であり、

チップの集積度に依存し、ベクトル数 N_v には依存しない。したがって、

$$T \propto N_v \quad (\text{式A1})$$

となる。

2) 斜めベクトルがある場合

$N_v \times P_D$ 本の斜めベクトルがあり ($0 \leq P_D \leq 1$)、斜めベクトルの x 座標軸に対する傾きの平均値を $\bar{\theta}$ 、degree が2の頂点数を N_{p2} とする。

式1~式5は以下のように変る。

$$N_p = 2 \times N_v - N_{p2} \quad (\text{式1'})$$

ここで、 $N_p \geq N_v$ であるから

$$N_{p2} \leq N_v \quad (\text{式6})$$

$$l_c = \{(2 \times N_v - N_{p2}) / D_p\}^{\frac{1}{2}} \quad (\text{式2'})$$

$$\begin{aligned} N_s &\leq \{\sqrt{2 \times N_v - N_{p2}} / (\overline{w_s} \times \sqrt{D_p})\} \\ &\quad + \{P_D \times \{1 - P_D \times (1 - \cos \bar{\theta})\} \times \sin \bar{\theta} \times \overline{l_v}^2 \times D_p \times N_v^2 / \{2 \times (2 \times N_v - N_{p2})\}\} \end{aligned} \quad (\text{式3'})$$

ここで、右辺オ2項はベクトル間の交点によるスリット増加数で、交点の x 座標が互に異なり、かつ、ベクトルの端点 x 座標とも異なるとした時の値である。

$$\overline{N_{vs}} = \{1 - P_D \times (1 - \cos \bar{\theta})\} \times \overline{l_v} \times \sqrt{D_p} \times N_v / \sqrt{2 \times N_v - N_{p2}} \quad (\text{式4'})$$

$$\begin{aligned} N_s \times \overline{N_{vs}} &\leq \{\overline{l_v} / \overline{w_s}\} \times \{1 - P_D \times (1 - \cos \bar{\theta})\} \times N_v \\ &\quad + P_D \times \{1 - P_D \times (1 - \cos \bar{\theta})\}^2 \times \sin \bar{\theta} \times \overline{l_v}^3 \times D_p^{1.5} \times N_v^3 / \{2 \times (2 \times N_v - N_{p2})^{1.5}\} \end{aligned} \quad (\text{式5'})$$

式5'の右辺は P_D の単調増加関数であり、 $P_D = 1$ 、 $N_{p2} = N_V$ (式6より) とおけば最大となるから、

$$\begin{aligned} N_S \times \overline{N_{VS}} &\leq \{ \overline{\ell_V} / \overline{w_S} \} \times \{ (\cos \bar{\theta}) \times N_V + (\sin \bar{\theta} \times \cos^2 \bar{\theta} \times \overline{\ell_V}^2 \times \overline{w_S} \times D_p^{1.5} / 2) \times N_V^{1.5} \} \\ &\div \{ \overline{\ell_V} / \overline{w_S} \} \times \{ 0(10^0) + 0(10^{-5}) \times \sqrt{N_V} \} \times N_V \\ &= \{ \overline{\ell_V} / \overline{w_S} \} \times f(N_V) \times N_V \end{aligned}$$

となる。 $N_V = 10^{10}$ とおくと $f(N_V) \div 2$ 、 $N_V = 10^8$ とおくと $f(N_V) \div 1.1$ 、 $N_V = 10^6$ とおくと $f(N_V) \div 1.01$ になる。したがって、 $N_V \leq 10^6$ ならば $f(N_V) \div 1$ とみなすことができ、

$$T \propto N_V$$

(式A2)

となる。

1)、2)の検討から、ベクトルスライス法の処理時間は入力ベクトル数にほぼ比例し、 10^6 ベクトル以下ならば斜めベクトルの影響で処理時間が長くなることはないと言える。

5. 結 言

マスクパターンデータを対象とした図形演算手法について、従来手法の整理と比較検討、および、高速化手法の提案を行なった。高速化手法の処理時間は、理論的に入力データ規模にほぼ比例することを述べたが、プログラム評価結果でも入力データ規模にほぼ比例することを確認済である。

[参考文献]

- [1] D.P. Treble: Dimensional Checking of M.O.S. L.S.I. Layouts; Int. Conf. on CAD IEE, p.7 (1972.4)
- [2] R.I. Stratford: Computer Aided Checking of Integrated Circuit Layout Constraints; Int. Conf. on CAD IEE, p.45 (1972.4)
- [3] M. Yamin: XYTOLR-A Computer Program for Integrated Circuit Mask Design Checkout; BSTJ, Vol. 51, No. 7, p.1581 (1972.9)
- [4] M. Yamin: Derivation of All Figures Formed by the Intersection of Generalized Polygons; BSTJ, Vol. 51, No. 7, p.1595 (1972.9)
- [5] D. G. Schweikert: A Computer-Generated Circuit Interpretation of Bipolar IC Masks; 著者より入手 (1972)
- [6] L.M. Rosenberg, et al.: CRITIC An Integrated Circuit Design Rule Checking Program; Proc. of DA Workshop, p.14 (1974.6)
- [7] C.L. Mitchell, et al.: MAP: A User-Controlled Automated Mask Analysis Program; Proc. of DA Workshop, p.107 (1974.6)
- [8] H.S. Baird, et al.: An Artwork Design Verification System; Proc. of DA Conf., p.414 (1975.6)
- [9] 小山田 哲治: L S I アートワークパターンの最小分割; 情報処理, Vol. 16, No. 7, p. 576 (1975.7)
- [10] B.W. Lindsay, et al.: Design Rule Checking and Analysis of IC Mask Designs; Proc. of DA Conf., p.301 (1976.6)
- [11] B.T. Preas, et al.: Automatic Circuit Analysis Based on Mask Information; Proc. of DA Conf., p.309 (1976.6)

- [12] I. Dobes, et al.: The Automatic Recognition of Silicon Gate Transistor Geometries: An LSI Design Aid Program; Proc. of DA Conf., p. 327 (1976. 6)
- [13] H.S. Baird: Fast Algorithm for LSI Artwork Analysis; Proc. of DA Conf., p. 303 (1977. 6)
- [14] 三橋 隆, 他: LSI のデザインルールチェック・プログラム; 情報処理, DA 31-1 (1977. 1)
- [15] K. Yoshida, et al.: A Layout Checking System for Large Scale Integrated Circuit; Proc. of DA Conf., p. 322 (1977. 6)
- [16] 小澤時典, 他: 論理接続チェック・システムの方式検討; 信学会半導体部門全大, No. 67 (1977. 8)
- [17] 江ロー彦, 他: LSI 接続チェックシステム I V S - 機能概要と基本構成 -; 信学会半導体・材料部門全大, No. 58 (1979. 10)
- [18] 吉村 寛, 他: マスクパターン接続チェック, 回路特性算出プログラム (MA I C A P); 信学会全大, No. S2-3 (1978. 3)
- [19] 杉山尚志, 他: 電子ビーム露光用データ処理システム; 信学会全大, No. S2-5 (1978. 3)
- [20] P. Wilcox, et al.: Design Rule Verification Based on One Dimensional Scans; Proc. of DA Conf., p. 285 (1978. 6)
- [21] M.A. Waters: A Data Base for Efficient VLSI Checking and Artwork Generation; Proc. of DA Conf., 発表せずコピーのみ配布 (1978. 6)
- [22] D. Alexander: A Technology Independent Design Rule Checker; 3rd USA-Japan Comp. Conf., p. 412 (1978. 10)
- [23] 吉田 純, 他: リニアIC マスク解析の一手法 (P A N A M A P - R) - 抵抗値算出プログラム -; 信学会, CST 78-125 (1979. 1)
- [24] 西出 雅, 他: LSI アートワークデータの図形論理演算処理の一手法; 信学会, CST 78-126 (1979. 1)
- [25] 大野日出夫, 他: 論理接続チェックシステム・P A L M S; 信学会全大, No. 409, 410 (1979. 3)
- [26] T. Akino, et al.: Circuit Simulation and Timing Verification Based on MOS/LSI Mask Information; Proc. of DA Conf., p. 88 (1979. 6)
- [27] C.R. McCaw: Unified Shapes Checker - A Checking Tool for LSI; Proc. of DA Conf., p. 81 (1979. 6)
- [28] C.S. Chang: LSI Layout Checking Using Bipolar Device Recognition Technique; Proc. of DA Conf., p. 95 (1979. 6)
- [29] P. Losleben, et al.: Topological Analysis for VLSI Circuits; Proc. of DA Conf., p. 461 (1979. 6)
- [30] S. Yamada, et al.: A Mask Pattern Analysis System for LSI (PAS-1); Proc. of ISCAS (1979. 7)