39. VLSI CAD への計算幾何学の応用†

鈴　木　則　久††

VLSI 用の CAD には計算機科学の種々の分野が応用されており，計算機科学応用の花と言ってよいだろう．関係ある分野としては，計算機アーキテクチャ，OS，データベース，プログラム言語，グラフィックス，アルゴリズム，数値計算，人工知能などがある．

ここでは，設計者が長方形の集合として VLSI 回路を表示し，それからトランジスタ回路を取り出したり，デザイン・ルールに合っているか検査する時に必要な，長方形間の交差部分を求めるアルゴリズムの動向を調べる．

1. まえがき

VLSI 用の **CAD** は計算機科学の種々の分野が応用されており，計算機科学応用の花と言ってよい．まず計算機アーキテクチャでは，CAD ワークステーション，特にグラフィックス用ハードの研究がある．OS ではワークステーション用分散 OS の研究，データベースでは，CAD 使用中に作成される種々のファイルの管理，プログラム言語では VLSI 記述言語，ソフトウェア工学ではシリコン・コンパイラ，グラフィックスでは VLSI レイアウト・システム，アルゴリズムではグラフィックス用とかレイアウト用データ構造とアルゴリズム，数値計算では回路シミュレーション，人工知能では知識工学 CAD である．

VLSI チップの設計段階では，幾何学的に設計されたチップから電気回路を摘出し，この電気回路を種々のシミュレータを通して，電気的・論理的特性を調べて設計の正誤をチェックするのが大変重要な仕事である．チップは一般に基準線に平行な 3 種（ポリシリコン，メタル，拡散）の長方形を重ねて設計してあり，

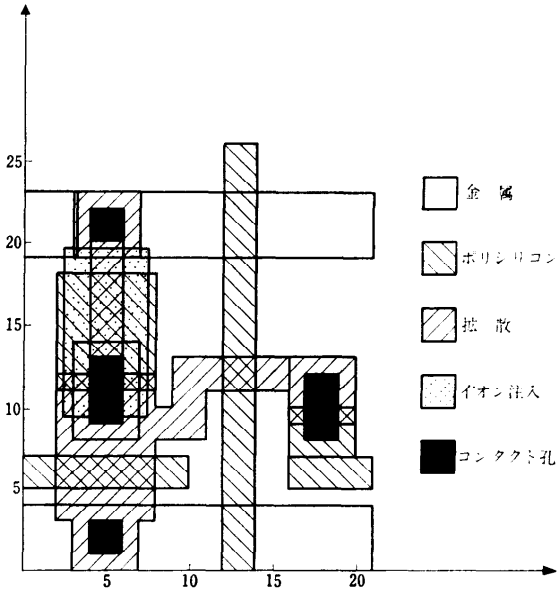

図-1　シフトレジスタ・セルのレイアウト[8]

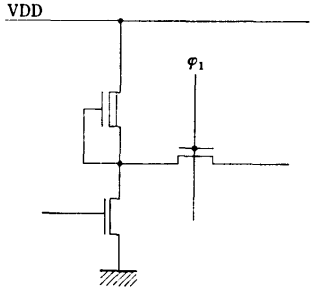

図-2　図-1 のレイアウトと等価な電気的回路

これらの交点が電気的に接点やトランジスタとなる[8]．よって，長方形の交点を求めれば電気回路が摘出できる．

例えば，**図-1**のように幾何学的に設計されたセルは，**図-2** にある電気回路を実現している．VLSI CAD では長方形の交点を高速に又なるべく少ない記憶領域で

† Application of Computational Geometry to VLSI CAD by Norihisa SUZUKI (Department of Mathematical Engineering, University of Tokyo).

†† 東京大学工学部計数工学科

求める必要がある．これは近年盛んになった計算幾何学の応用である．このサーベイでは，長方形の交点を求めるアルゴリズムの動向を調べる．

2. 長方形の交点を求めるアルゴリズム

2次元座標軸のある平面で，点 $(0,0)$，$(R,0)$，$(0,R)$，(R,R) により決められる正方形を考える．この中に，座標軸に平行な長方形が n 個置かれている．これらの長方形の中の任意の2つの長方形が交わっているかどうか調べるのが目的である．

このアルゴリズムの入力は，n 個の長方形を決定する2点の座標の集合であり，出力は交点である S 個の長方形を決定する2点の座標の集合である．

2.1 歴史的背景

この問題は単純にやれば $O(n^2)$ の時間がかかる．Bentley & Wood[1] は時間 $O(n \log n)$ で空間 $O(n \log n)$ のアルゴリズムを作った．その後，次々と時間 $O(n \log n)$ で空間 $O(n)$ のアルゴリズムが，McCreight[2]，Imai & Asano[3]，Edelsbrunner et al[4]，McCreight[5] によって作られた．ここでは McCreight[5] の作り出した優先探索木 (Priority Search tree) を基にしたアルゴリズムを中心に紹介する．

2.2 優先探索木

McCreight[5] は**優先探索木** (Priority Search tree) というデータ構造を考え出し，これを使って長方形の交差部分を求めるアルゴリズムを作った．優先探索木には基数型とバランス型があるが，ここでは基数型を基にして話を進める．

(a) 優先探索木での操作

優先探索木を抽象データ型を使って表わし，行うことの出来る操作を定義する．

ドメイン D は，$0, 1, \cdots, k\text{-}1$ までの整数よりなるペアー $[x, y]$ である．操作としては次の6つが行える．

- Insert Pair (x, y): $[x, y]$ を D に挿入する．
- Delete Pair (x, y): D から $[x, y]$ を除去する．
- Min X In Rectangle $(x\phi, x1, y1)$: 与えられた $x\phi, x1, y1$ に対し，D の要素 $[x, y]$ で $x\phi \leq x \leq x1$，$y \leq y1$ を満たしてかつ x が最小となるペアーを求める．
- Max X In Rectangle $(x\phi, x1, y1)$: $x\phi, x1, y1$ が与えられたとき，D の要素 $[x, y]$ で $x\phi \leq x \leq x1$，$y \leq y1$ を満たしてかつ x が最大となるペアーを求める．

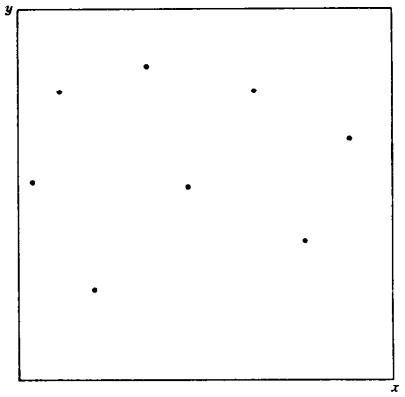

図-3 D におけるペアー $[x, y]$ の2次元表現

- Min Y In X Range $(x\phi, x1)$: $x\phi$ と $x1$ が与えられたとき，D の要素 $[x, y]$ で $x\phi \leq x \leq x1$ を満たすもののうち y が最小のものを求める．
- Enumerate Rectangle $(x\phi, x1, y1)$: $x\phi, x1, y1$ が与えられたとき，D の中で $x\phi \leq x \leq x1$ と $y \leq y1$ を満たすすべての要素 $[x, y]$ を求める．

(b) アルゴリズム作成の方針

まず D に制限をもうけて，x 座標の同じペアーは存在しないとする．問題によっては x 座標の等しいペアーを扱わなければならないこともある．その時は，ある関数 F を作り，$[x, y]$ を $[F(x, y), y]$ に写影することによって x 座標が等しくないようにする．

D を2次元平面上に表わすと **図-3** のようになる．y について最小である判定と，y の値がある値より小さい範囲で，x についての範囲判定を行うのであるから，まず y 座標の最小のものを選んでそれを2進木の根におく．次に x 座標をま二つに切り，各々の部分領域で y が最小のものを部分木の根とする．このプロセスを繰り返していくと，最後には，x 座標での幅が1の領域が出来る．D の中で，x 座標が同じ点は2つ以上存在しなかったので，x 座標の幅が1の領域にはたかだか1つの点しか存在しないので，$\log k$ ステップで止まる．**図-4** の点の集合からは，**図-5** の優先探索木が得られる．

(c) データ構造実現法

まず優先探索木の各ノードは PASCAL[6] で書くと次のように表わされる．

```
TYPE
  Key Range=First Key..Last Key;
  Pair=RECORD x,y: Key Range END;
  RPST Ptr=↑RPST;
```

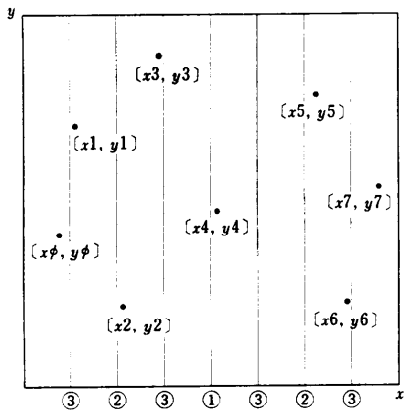

図-4 基数探索を行う為に X 座標で分割する.

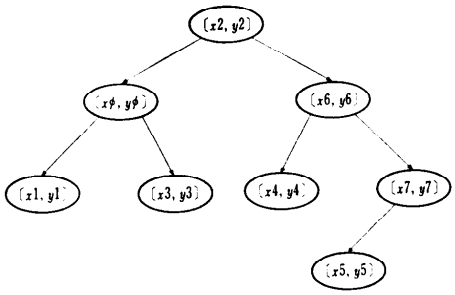

図-5 図-4 の点を表現する優先探索木

```
RPST=RECORD
  p: Pair;
  left, right: RPST Ptr;
  END;
```

まず優先探索木では，Dの要素はただ1つのノードの p 項にのみ現われる．よって，n 点からなるDを表現するには $O(n)$ の領域が必要である．また，優先探索木は次の2つの不変性を満たす．

第一の不変性は y 項に関するもので，今 t がノードを指している時，t.lcft が NIL でなければ，$t.p.y \leq t.\text{left} \uparrow.p.y$ であり，t.right が NIL でなければ，$t.p.y \leq t.\text{right} \uparrow.p.y$ である．

第二の不変性は x 項に関するもので，あるノード t に対して，$t.p.x$ は [lower..upper) の範囲内の値を取るとすると，t.left が NIL でなければ，$t.\text{left}.p.x$ は [lower..floor ((lower+upper)/2)) の範囲内の値を取り，t.right が NIL でなければ．$t.\text{right}.p.x$ は [floor ((lower+upper)/2)..upper) の範囲内の値を取る．

(d) アルゴリズム

Insert Pair では，まず根にあるペアーの y と新しく挿入するペアーの y を比べる．もしも根の y の方が小さければ，新しいペアーの x の値により，左か右の部分木に挿入される．新しいペアーの y の方が小さければ，根にあるペアーを取り出し，根には新しいペアーを入れ，取り出したペアーは x の値により，左か右の部分木に挿入する．

Delete Pair では，まず除去しようとするペアーを探す．見つかったら，そのペアーを除くので木に穴が1つあくので，それを2つの子のうちで y 座標の小さな方で埋める．すると穴は1つ下がるので，下の穴を埋めるのを前と同じ操作を繰り返して達成する．

Min X In Rectangle では，まず根の y の値，$t.p.y$ が範囲内かどうか調べる．もし範囲外なら，この木には求めるペアーはない．範囲内ならば，まず左部分木の中で解があるか調べ，無いときにのみ，右部分木の中で解を求める．最後にこうして求めた部分木での解と根のペアーとを比べ，x の小さい方が解である．

Min Y In X Range では，根の y の値が木の中のペアーの y よりも小さいので，根の x が範囲内ならばそれが解である．そうでない時は，左部分木と右部分木の解の小さい方が解である．

Enumerate Rectangle は depth-first 探索で可能性のあるノードは皆見る．

(e) 実行時間の解析

Insert Pair と Delete Pair は木の1つの枝しか探さないので，$O(\log k)$ の実行時間しかかからない．

その他の解析にはもっと複雑な考察が必要だが，それは原論文を参照されたい．

2.3 優先探索木を使って長方形の交点を求めるアルゴリズム

まず優先探索木を使って，1次元の区間の重なりを求めるアルゴリズムが作れることを示す．これには，D の要素 $[x, y]$ で区間 $[y..x]$ を表わす．今，区間 $[u.v]$ と $[y..x]$ が重なりを持つ必要十分条件は，$u \leq x \leq$ Last Key でかつ First Key $\leq y \leq v$ である．すなわち，Enumerate Rectangle (u, Last Key, v) が求める解を与える．これにはデータ構造に $O(n)$，区間の挿入・削除に $O(\log k)$，数え上げるのに $O(\log k + S)$ かかる．

長方形の重なりを見つけるには，**平面走査法**[7] を用いる．この方法では，平面上を水平線を下から上へ移動し，水平線が長方形を切断して出来る断面を見て演算を実行する．

今考えている平行な長方形では，断面は重なりを持つ区間の集合となる．また区間の重なりが，長方形の交差部分を示す．走査線によって出来る断面は，長方形の上と下の辺を切る時のみ変化するので，たかだか長方形の数の2倍だけの断面において，区間の重なりを調べればよい．

具体的には，走査線が上に動いて行って新しく長方形にぶつかると，下辺に対応する区間と，今までに存在する区間の重なり合わせを数え上げる．この重なり合わせが，交差部分を示す．そして，この区間を区間の集合に挿入する．走査線が長方形の上端から落ちる時は，上辺が区間の集合より除去される．

2.4 例

図-6 の例を考えて見よう．

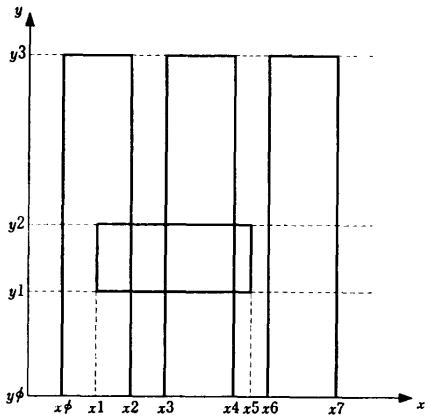

図-6 平均走査法により交点を求める．

走査線が$y1$に来た時には，区間の集合は$[x\phi..x2]$, $[x3..x4]$, $[x6..x7]$ である．$[x1..x5]$ との重なりを調べるには，Enumerate Rectangle ($x1$, Lastkey, $x5$) で求まる区間は $[x\phi..x2]$ と $[x3..x4]$ である．

3. 課 題

ここで挙げたアルゴリズムは，オフラインアルゴリズムである．すなわち，設計者がレイアウトをすべて終えてから，回路を摘出し，シミュレーションするのに使われる．しかしながら，これからの CAD システムではもっと高速なリスポンスが要求されよう．設計者が一つの長方形を置くと同時に，回路は摘出され，シミュレーションが行われ，回路の論理値が画面ですぐ見られるようになることが好ましい．

この為に必要なアルゴリズムは，オンラインアルゴリズムである．オンラインアルゴリズムのことについては，まだ誰も研究していないし，何の結果も出ていないから，どのようなデータ構造が良いか判っていない．

4. あ と が き

ここでは，VLSI CAD で最も重要な部分の一つである回路摘出の問題を取り上げ，その解法アルゴリズムの動向と，その中の最良複雑さを持つ Mc-Creight の優先探索木を紹介した．これで，$O(n \log n)$ の時間と，$O(n)$ の空間で，オフライン回路摘出が行える．これからの研究課題は，オンライン回路摘出が出来るようになることである．

編集者，査読委員の方の適切なコメントに感謝する．

参 考 文 献

1) Bentley, J.L. and Wood, D.: *An Optimal Worst Case Algorithm for Reporting Intersections of Rectangles. IEEE Trans. Comp.* C-29, pp. 571-577 (1980).
2) McCreight, E.: *Efficient Algorithms for Enumerating Intersecting Intervals and Rectangles, Xerox Palo Alto Research Centers Technical Report CSL*-80-9, Xerox Corporation, Palo Alto (1980).
3) Imai, H. and Asano, T.: *Finding the Connected Components and a Maximum Clique of an Intersection Graph of Rectangles in the Plane*, Dept. of Math. Eng., U. of Tokyo (1981) (to appear in *Journal of Algorithms*).
4) Edelsbrunner, H. et al.: *Connected Components of Orthogonal Geometrical Objects*, F 72, Institut für Informationsrerarbeitung, (1981).
5) McCreight, E.: *Priority Search Trees, Xerox Palo Alto Research Centers Technical Report CSL*-81-5, Xerox Corporation, Palo Alto (1982).
6) Jensen, K. and Wirth, N.: *PASCAL User Manual and Report*, Springer-Verlag, Berlin (1978).
7) Shamos, M.I. and Hoey, D.: *Geometric Intersection Problems, 17th Annual Symposium on Foundations of Computer Science*, pp. 208-215, IEEE (1975).
8) Mead, C. and Conway, L.: *Introduction to VLSI Systems*, Addison-Wesley, Reading, Mass. (1980). なお VLSI CADの参考書としては，
9) Rice, Rex, ed., *Tutorial VLSI Support Technologies: Computer-Aided Design, Testing, and Packaging*, IEEE Computer Society Press, New York (1982).

(昭和57年12月1日受付)