

VLSIレイアウトシステム(VILLA)用  
自動配線プログラムについて  
山本一郎, 森 裕子, 小池 豊, 荘司公克, 安藤 宏  
( 沖 電 気 工 業 株 )

1.はじめに

L S Iの集積度向上に伴い, L S Iのレイアウト設計は年々困難になっている。特にカスタムL S Iでは, 短い設計期間で誤りのないレイアウト設計が必須であり, ゲートアレーやスタンダードセルのセミカスタムL S I自動設計システムが開発され, 実用化されてきた。しかし, これらはブロックの形状やブロックの端子位置等に制限を設け, チップサイズは犠牲にして自動設計するもので, チップサイズを最小化しようとするフルカスタムには向かない。我々は, 今回ブロックの形状, 端子位置, 配置法等の制限の少ない階層形ビルディングブロックレイアウトシステム(VILLA)〔1〕を開発した。本論文では, VILLAで用いた自動配線プログラムについて述べる。

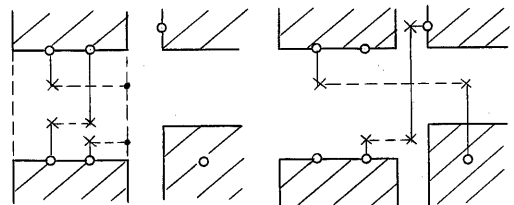
2. 基本アルゴリズム

配線アルゴリズムとしては,

- ① L/S (ラインサーチ) 法〔2〕
- ② CA (チャンネルアサイメント) 法〔3〕
- ③ 迷路法〔4〕

が知られている。これらのうち, 迷路法は使用する記憶容量の点で問題があるため, 基本アルゴリズムの候補から外した。チャンネルアサイメント法は, 配線領域をチャンネルと呼ばれる矩形に分割し, 配線を各チャンネル内の独立した問題として取り扱う手法である。L/S法は, ネット単位に結線を試みる手法で, 結線されたネットの配線経路は, 以後のネットに対する障害物となる。

L/S法とCA法を比較すると, 次のようになる。



(a) CA法 (b) L/S法

図1. CA法とL/S法

(1)レイアウトモデル

CA法では, 矩形のチャンネルを構築する必要があるため, ブロック外形状や配置法に制約が付く。またチャンネル内配線を行う時点で, その内部を通過する全系列の端子をチャンネル周辺に設定する必要があるため, 端子位置に制約が付く。これに対し, L/S法では, 下位ブロック内部も一般配線領域として取り扱えるため, CA法のような制約が少ない。また, L/S法は多層配線に容易に拡張可能であるが, CA法では3層以上の配線層を同時に取り扱うのは困難である。このように, L/S法はCA法よりも自由度の高いレイアウトモデルが実現できる。

(2)処理時間

CA法では, 各チャンネル内での配線を別個の問題として取り扱えるため, L S Iの大規模化による影響は少ない。これに対し, L/S法では, 最悪の場合, 系列数と格子点数の積に比例する。

(3)結線率

CA法は, 各チャンネル内の配線時に, 必要トラック数に応じて上下(左右)の下位ブロックを移動させ, チャンネルを拡張することで, 結線率=100%が保証できる。しかし, その代償とし

〔1〕

て、配線終了時までブロックの外形が決定できないことになる。一方、L/S法では、早い段階で引かれた配線経路が、配線順序の遅い系列の障害となり、未結線となることがある。このため、100%は保証されない。

#### (4) インタラクティブ性

人手で配線を行うとき、人間は種々の要素を同時に判断することができるが、基本的な作業の単位はL/S法と同じく系列である。したがって、インタラクティブ配線を実現する場合は、L/S法の方が設計者の思考を反映し易いシステムが構築できる。

我々は、L/S法におけるレイアウトモデルの自由度の高さ、および、インタラクティブ性に着目し、これを採用することにした。

この結果、

- (i) 結線率の問題
- (ii) 処理時間の問題

が、プログラム開発上での課題となった。

### 3. 課題解決法

2章で述べたL/S法の2つの課題を解決するため、我々は次の対策を取った。

#### (1) 結線率の問題

- 広く一般的に用いられている手法であるが、配線を概略配線と詳細配線の2段階に分割することにより、局所的な配線の集中をさける。
- すでに結線済みの他系列の配線を、ダイナミックに変更しながら着目系列の結線を行う。

#### (2) 処理時間の問題

L/S法における処理の大多数は、探索に用いる仮線分の発生に費やされる。このため、経路探索に必要な仮線分が少ないほど、処理時間が少なくてすむ。そこで、次の手法を取った。

- 概略配線により系列ごとに探索域を設定する。この探索域は閉図形で表現されており、詳細配線はその内部のみを経路探索し、余分な領域は探索しない。
- 新たに考察した新L/Sアルゴリズム（先行探索）を用いることにより不要な仮線分を減少させる。

### 4. 概略配線

概略配線の目的は、混雑度の平滑化と、詳細配線が経路探索を行う領域（探索域）の限定にある。

#### (1) 処理手順（図2）

- ① 設計対象のブロックおよびその下位ブロックの外形をもとに、配線領域をチャンネルと呼ぶ領域に分割する。
- ② 各系列の探索域を、線長、折れ曲り、混雑度を評価して、チャンネルの並びとして決定する。

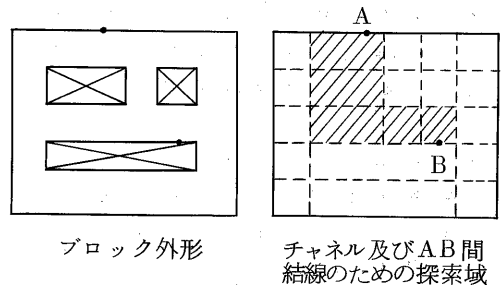


図2. 概略配線

#### (2) 再概略配線

概略配線の最大の問題は、混雑度の見積り誤差である。これは、概略配線の評価単位がチャンネルであるため、局所的な未使用部分や混雑は評価できないことが原因である。あるチャンネルの混雑度を過少に評価すると、そのチャンネルに割り付けられた系列に未結線が発生する。逆に過大に評価することは、配線領域の無駄使い（チップサイズの増加）につながる。

そこで、詳細配線で未結線となった

系列は、概略配線で探索域の再設定（再概略配線）を行うようにした。この再概略配線では、詳細配線の結果を混雑度の算出に反映させることにより、未結線系列を過少評価チャンネルから、過大評価チャンネルへ再割り付けする処理を行う。詳細配線は、これらの再設定された未結線系列に対し、結線を試みる。

## 5. 詳細配線

詳細配線プログラムでは、概略配線より与えられた探索域内で、設計規則を満足させながら、実際の結線を行なう。本章では、そのアルゴリズムについて述べる。

### 5-1. 従来のL/Sアルゴリズム

従来のL/S法アルゴリズムは、次のステップからなる。ここに記されている仮線分とは、経路を探索するために、配線格子上を上下または左右方向に、その配線層の障害物に達するまで伸ばした線分である。伸ばす方向は、配線層ごとにX、Yと交互に設定されており、これを走方向と呼ぶ。

#### ・従来のL/S法（図3）

- ①  $0 \rightarrow I$
- ② 端子  $T_1$  より、その端子の配線層の走方向に仮線分を発生する。これを  $T_1$  側レベル0 仮線分と呼ぶ。もし、相手端子  $T_2$  と交差したならば、経路が求まったとして終了。
- ③ 端子  $T_2$  より仮線分を発生させる。これを  $T_2$  側レベル0 仮線分と呼ぶ。もし、 $T_1$  側レベル0 仮線分と交差したならば、経路が求まったとして終了。
- ④  $I + 1 \rightarrow I$
- ⑤  $T_1$  側レベル  $(I - 1)$  仮線分上に  $VIA$  を取り、隣接配線層上に仮線分を発生させる。これらの仮線分を  $T_1$  側  $I$  レベル仮線分と呼

ぶ。もし、 $T_2$  側レベル  $(I - 1)$  仮線分と交差したならば、経路が求まったとして終了。一本も発生しないならば、未結線として終了。

- ⑥  $T_2$  側についても、ステップ⑤と同様の処理を行う。
- ⑦ ステップ④へ

上記処理ステップ中に記されている『レベル』を定義すると、次のようになる。

#### ・定義1

レベル  $I$  とは、端子から到達するのに必要な折れ曲りの数が、 $I$  回である仮線分の集合である。

また、レベルには次の性質がある。

#### ・性質1

レベル  $I$  は、レベル  $(I - 1)$  から1回折れ曲ることにより到達できる仮線分である。

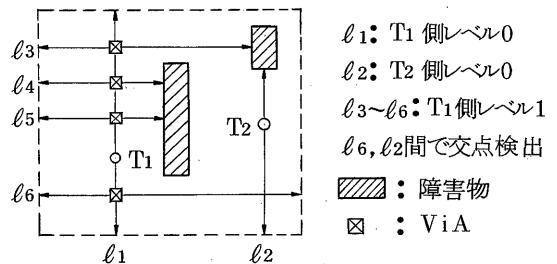


図3. 従来のラインサーチアルゴリズム

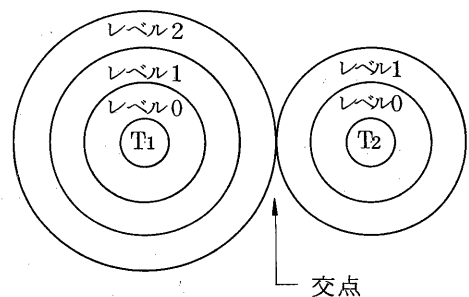


図4. 端子  $T_1, T_2$  より発生するレベルの概念

・性質 2

レベル I の仮線分と交差する可能性がある仮線分は、相手端子の最上位レベルの仮線分である。(図 4)

これらの性質があるため、L/S法では、経路は最少折れ曲りで求まる。

5-2. 屈曲探索域問題

4章で述べたように、VILLAの詳細配線プログラムは、図5のような屈曲した探索域を扱う。ここに、図5中の「折れ曲り領域」とは、屈曲探索域の折れ曲った部分であり、「区分」は、端子または折れ曲り領域を両端とする領域を示す。ただし、折れ曲り領域は、接続する2区分の両方に属するとする。

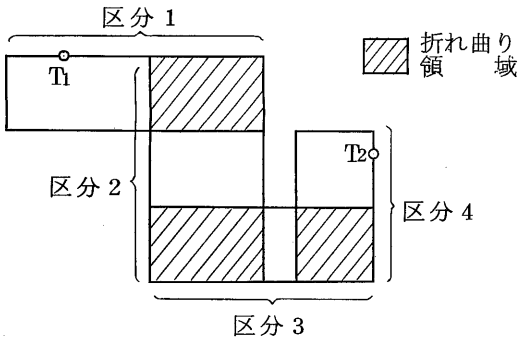


図5. VILLA詳細配線の探索域 (屈曲探索域)

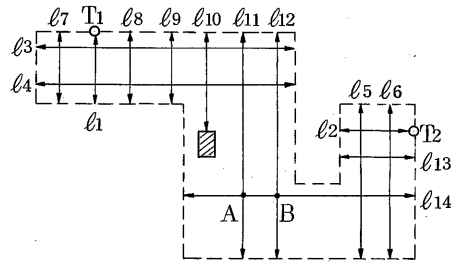
図6は、従来のL/S法を屈曲探索域に用いた例である。この図のT1側レベル2に着目すると、結線に有効と思えない仮線分(劣性仮線分)  $l_7, l_8, l_9$  を発生させていることがわかる。逆に結線に有効な仮線分は、折れ曲り領域から発生した  $l_{10}, l_{11}, l_{12}$  である。

このように、屈曲探索域に従来のL/S法を適用すると、劣性仮線分が大量に発生する。しかし、これらの劣性仮線分を発生させないとすると、図7のような、解があるのに求まらないケ

ースが発生する。よって、L/S法におけるレベルの概念を変え、劣性仮線分を発生させるレベルを遅らせ、折れ曲り領域から発生する仮線分を早いレベルとするのが望ましい。これを実現したのが、本論文で述べる先行探索である。

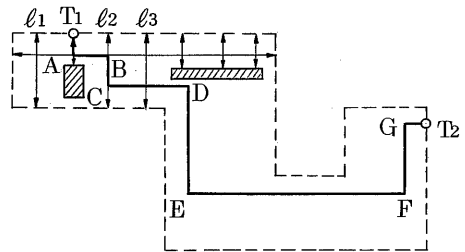
5-3. 先行探索におけるレベル

5-2で述べたように、屈曲探索域では、折れ曲り領域から発生する仮線分を、早いレベルで発生させるのが望ましい。これを実現するためには、配



- T1側レベル 0 :  $l_1$
- T2側レベル 0 :  $l_2$
- T1側レベル 1 :  $l_3, l_4$
- T2側レベル 1 :  $l_5, l_6$
- T1側レベル 2 :  $l_7 \sim l_{12}$
- T2側レベル 2 :  $l_{13}, l_{14}$
- 交点 : A B

図6. 屈曲探索域における従来のラインサーチ



劣性仮線分  $l_1, l_2, l_3$  を作成しないと  $T_1-A-B-C-D-E-F-G-T_2$  なる解が求まらない。

図7. 劣性仮線分の必要な場合

線は折れ曲り領域内で必ず1回は折れ曲るので、各折れ曲り領域内での最初の折れ曲りは絶対に必要なものであるとして、レベルの上昇にカウントしなければよい。そこで、先行探索では、レベルの定義を次のように変えた。

• 定義2

レベルIとは、端子より到達するのに必要な折れ曲りのうち、絶対折れ曲りを除いたものの数が、I回である仮線分の集合である。ただし、絶対折れ曲りとは、各折れ曲り領域内での最初の折れ曲り、および、端子から出る方向が探索域の進行方向に垂直な場合の最初の折れ曲りを意味する。

これを、図8を用いて説明すると次のようになる。端子T1から、仮線分 $l$ に達するまでに必要な折れ曲りは、A、B、C、D、E、Fの6回である。これらのうち、B、C、Fは折れ曲り領域内での最初の折れ曲りであるので、絶対折れ曲りである。また、端子T1から出る方向は探索域の進行方向に垂直であるため、Aも絶対折れ曲りである。よって、残りはD、Eであるから、仮線分 $l$ はレベル2の仮線分である。なお、従来のL/S法で行なえば、 $l$ はレベル6とかなり発生が遅くなる。

レベルの定義をこのように変更しても、従来のL/S法の持つ性質2は成

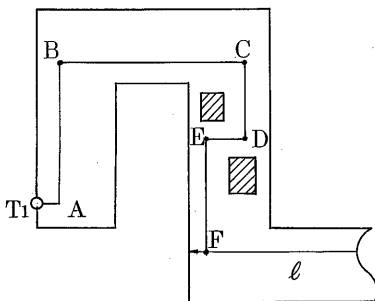


図8. 先行探索のレベル

立する。また、先行探索のレベルにおいて、性質1と同等な次の性質があることがわかる。

• 性質3

レベルIは、レベル(I-1)から1回折れ曲ることによって到達可能な仮線分、および、その仮線分から絶対折れ曲りをくり返すことにより到達可能な仮線分である。

結線に必要な絶対折れ曲りの回数は探索域の形状により定まるものであり一定である。よって、性質2、3より先行探索においても、経路は最少折れ曲りで求まる。

5-4. 先行探索基本アルゴリズム

先行探索の基本アルゴリズムは、レベルの定義が変わったのみで、5-1で述べた従来のL/S法と同一である。そこで、その処理ステップの記述は省略するが、5-1のL/S法処理ステップのうち、②③⑤⑥に相当する着目レベル内の仮線分を発生させる処理については、(1)で詳細に述べる。

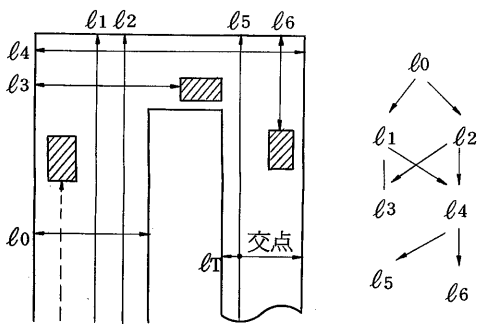
(1) レベル内仮線分発生処理

レベル内仮線分発生処理とは、レベル(I-1)を親として、レベルIの仮線分を発生させる処理、および、発生させた仮線分と相手端子側仮線分との交差チェックを行う処理である。いま、レベルIの仮線分を、探索域の進行方向に並行なもの(進行仮線分)と、垂直なもの(垂直仮線分)に分けて考える。進行仮線分は障害物がない限り次の折れ曲り領域に到達し、その仮線分を折れ曲り領域内で折り曲て発生させる仮線分もまた、同じレベルIに属し、かつ、進行仮線分である。これに対し、垂直仮線分は折れ曲り領域に達することはないので、従来のL/S法に比べ、特殊な処理をする必要はない。

本節では、進行探索線を発生させる場合について、説明する。

☆進行仮線分発生法

図9-aは、レベル(I-1)の仮線分 $l_0$ から、レベルIの仮線分 $l_1 \sim l_8$ を発生させた例である。これらの仮線分の関係を図9-bとなる。ここに、グラフのノード間の連結は、仮線分の親子関係を示している。このグラフにおいて、深さが深い仮線分ほど相手端子に近くなるため、進行仮線分を発生させる場合は、深さを優先する順序で発生させるようにした。次に、そのステップを、図9を例に用いて記す。



(a) 進行仮線分発生例 (b) 仮線分の関係

図9. 進行仮線分

・処理例

- ① レベル(I-1)の仮線分 $l_0$ から、仮線分を1本発生させる。
- ② 発生した仮線分 $l_1$ は折れ曲り領域に達したので、 $l_1$ を親として折れ曲り領域より仮線分を1本発生させる。
- ③ 発生した仮線分 $l_3$ は次の折れ曲り領域の前で停まったので、 $l_1$ を親として仮線分をもう1本発生させる。
- ④ 発生した仮線分 $l_4$ は次の折れ曲り領域に達したので、 $l_4$ を親として折れ曲り領域より仮線分を1

本発生させる。

- ⑤ 発生した仮線分 $l_5$ は、相手端子側仮線分 $l_T$ との交差があるので、配線経路が見つかったとして終了。

(2) 交差チェック対象の限定

先行探索には先に述べた性質2があるため、交点の有無を調べるには、相手端子側の最上位レベルの仮線分に対してのみ調べればよい。この点では、5-1で述べた従来のL/S法も同じであるが、先行探索では、仮線分を区分を単位に管理することにより、交差チェックの相手をレベルと区分で限定することで、一層の高速化を図った。

5-5 既配線押しのけ処理

詳細配線プログラムでは、結線済み他系列の配線による未結線を少なくするために、「既配線押しのけ処理」を行なっている。この処理には、次の2通りがある。

(1) ピンブロック解除 (図10)

端子から出る仮線分が、既に存在する他系列の配線にぶつかる場合は、その配線を未使用領域に平行移動することにより、結線可能とする。

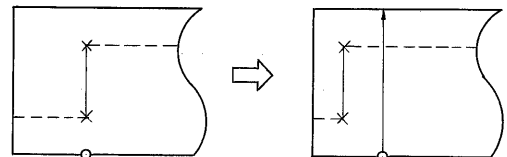


図10. ピンブロック解除

(2) チャネルブロック解除

5-5で述べた進行探索線を発生させる処理において、仮線分が1本も次の折れ曲り領域に達しない場合は、既に存在する他系列の配線を未使用領域に平行移動することにより、この状態を解除する。

6. 結果

本自動配線プログラムを，CMOS 20KG級のVLSIに適用した例を表1に示す。ここに，ケースA，Bは中間階層のブロックであり，ケースCは，ケースB相当の中間階層ブロック4個とRAMからなる最上位階層ブロックである。また，実行方法はそれぞれ次のとおりである。

- a : 既配線押しのけ処理を用いずに，(概略配線+詳細配線)を1回行う。
- b : 既配線押しのけ処理を用いて，(概略配線+詳細配線)を1回行う。
- c : 既配線押しのけ処理を用いて，(概略配線+詳細配線)を2回

表1より，「既配線押しのけ処理」，「再概略配線」に効果があることがわかる。また，CPU時間に関しては，従来のL/S法に比べ1/5~1/10に短縮されたと思われる。(実施例は，約7MIPSの汎用計算機を使用)

7. おわりに

階層手法を用いたVLSIレイアウトシステム(VILLA)の自動配線プログラムについて説明し，それに用いた新探索アルゴリズムを提案した。簡単化のため，本論文では走方向は

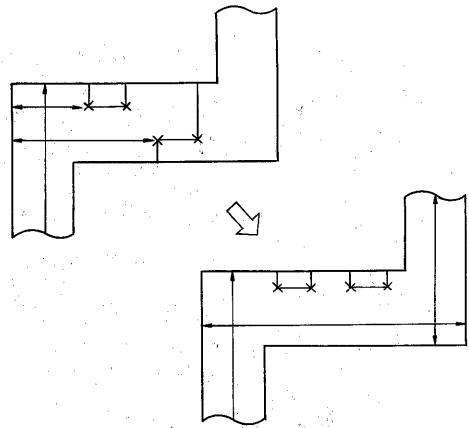


図11. チャネルブロック解除

配線層ごとにX，Yと交互に設定するとしたが，実際のレイアウトでは，電氣的に好ましい配線層を多く使用するために，配線領域を分割し，各部分領域ごとに走方向を設定する手法を用いた。

今後は，より一層の高結線率化と高速化をめざすのみならず，設計者の指示で会話型に配線する「インタラクティブ配線」に発展させてゆきたい。

表1. 実施例

ケース	下位ブロック数	2端子間数	未結線本数 (CPU時間)		
			実行方法 a	実行方法 b	実行方法 c
A	119	548	14 (8m)	11 (9m)	4 (12m)
B	1196	3688	12 (56m)	10 (64m)	6 (68m)
C	5	1020	14 (8m)	14 (10m)	0 (23m)

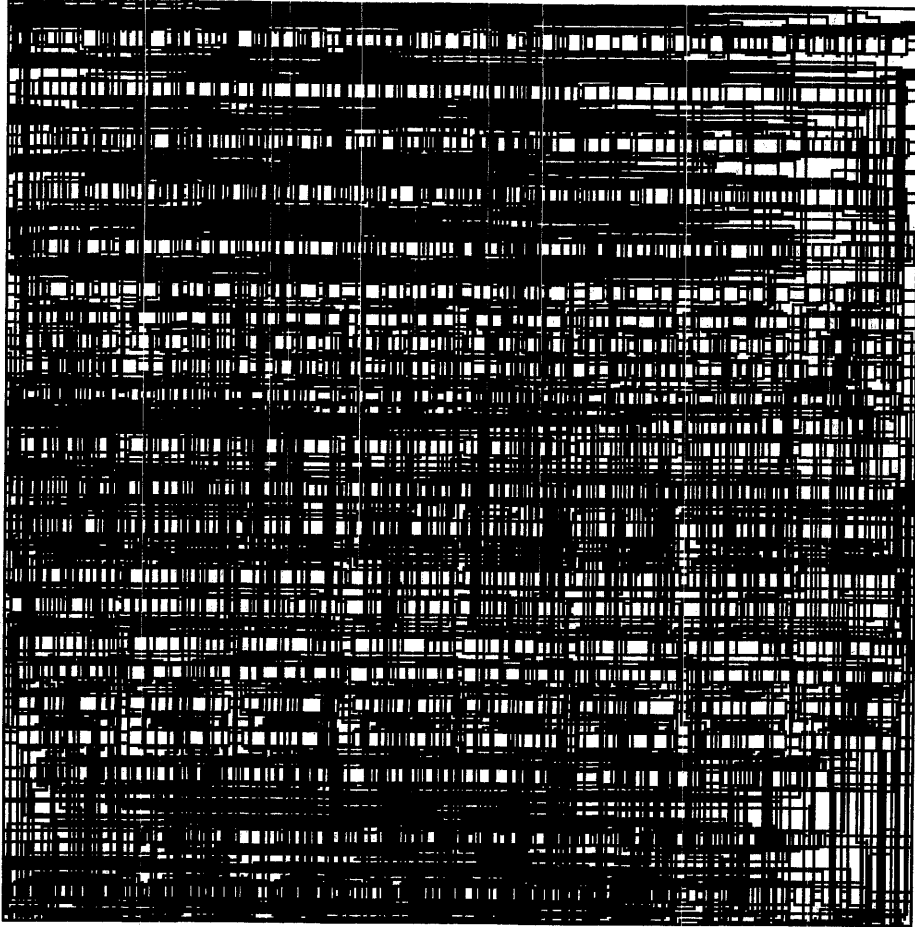


図12. 実施例 (ケースB)

【参考文献】

- [1] 荘司, 他「VLSIレイアウトシステム (VILLA) の構成」  
情報処理学会第29回全国大会
- [2] Mikami, K. and Tabuchi, K.: a Computer Program for Optimal Routing of Printed Circuit Conductors, IFIP Congress68, pp. 1475-1478 (1968).
- [3] Hashimoto, A. and Stevens, J.: Wiring Routing by Optimizing Channel Assignment within Large Apertures, Proc. 8th Design Automation Workshop, pp. 155-169 (1971).
- [4] Lee, C. Y.: An Algorithm for Path Connections and Its Applications, IRE Trans. Electron. Comput., Vol. EC-10, pp. 346-365 (1961)