

# 物理設計予測に向けた 多端子を扱う超高速配線法の研究

A Very Fast Multi-terminal Net Routing Method for Physical Design Prediction

岡村 歩<sup>1)</sup> 藤井 良弥<sup>2)</sup> 豊永 昌彦<sup>3)</sup>

Ayumu Okamura<sup>1)</sup> Yoshiya Fujii<sup>2)</sup> Masahiko Toyonaga<sup>3)</sup>

1) 高知大学理学部 2) 高知大学大学院理学専攻 3) 高知大学 情報講座

Information Science Division, Faculty of Science, Kochi University

## あらまし

本研究は、詳細配線を実施する前に、配線後の回路性能を測定するための超高速配線法の提案とその効果の検証をおこなったものである。提案する多端子を扱う超高速配線法は、2端子ネットをL型、逆L型の単純な配線生成し、他ネットとの配線重複が最少となる配線経路を求める。多端子ネットの扱いは、まず2端子へ分割し配線生成を行い、同一ネットの2端子配線が多重評価されるのを避けるためグリッドマップを使う。多端子ネットを2端子ペアへの分割割り付けは、多端子の迷路配線形状に近い割り付けをPrim's MST Algorithmにより行う。

高速化と配線形状を調べる実験において、実回路8bitマイコンと比較したところ、従来配線法に比べて処理時間が約315倍で、同様な配線形状が得られた。

**キーワード:** QTAT, 多端子配線, 迷路配線

## 1. はじめに

情報化社会の発展で、電子機器の高性能化、低電力化、小型化が市場より求められている。

これを支えるため電子機器の核心部品であるVLSI (Very Large Scale Integrated: 超大型集積回路)は、高性能化、低電力化、小型化が求められ、VLSI製造プロセスの微細化が進んできた。その理由は、VLSI製造を微細化することによりVLSIの回路部品(抵抗、コンデンサ、トランジスタなど)の搭載数を増大でき、信号配線長を短くできるので信号伝搬が高速化され高性能になること、さらに回路サイズの縮小で消費電力が削減できるからである。

このVLSI製造微細化とそれに伴うVLSI性能の向上は「集積回路に搭載されるトランジスタ数は約18ヶ月で2倍になる」という「ムーアの法則」としてよく知られている。2011年現在では、VLSIの代表であるCPU (Central Processing Unit: 中央処理装置)のトランジスタ数が20億個にも達している[1]。VLSI製造微細化に伴い回路規模が増大するため、設計技術では回路設計期間の短縮に向けた高度な自動設計技術と、VLSI性能を引き出す高速回路設計技術への取り組みが行われている。

極度に微細化が進んだ近年のVLSI製造では、レイアウト設計後の配線間容量がクロストークノイズの原因となり信号誤動作を生じる。また配線抵抗の増加で長距離配線の信号遅延が長くなり動作不良となる場合が多くなる。1 $\mu$ m以上のサイズでは回路動作は、トランジスタ性能で決めることができた。しかし、極微細化に伴いトランジスタ性能向上に対して配線抵抗の信号遅延が大きくなり、さらに動作周波数が高く隣接配線間の信号干渉が大きくなり物理設計の考慮が必須となってきた。

レイアウト設計前の機能設計では、シリコン回路の配線や素子が決っていないため、直接物理設計を見積もることはできない。そのため機能設計では、既存の類似する回路の結果を流用した見積もりが行われる。

また、シリコン回路の配線や素子を決める論理設計でも、配線形状や素子位置が決っていないため、直接物理設計を見積もることはできない。そのため論理設計では、既存の類似する回路の結果から得られた配線長情報でタイミング見積もる。

この論理設計で用いる配線長情報は、ワイヤ・ロード・モデルと呼ばれ、信号(ネット)の出力端子をもつ素子の駆動能力と入力端子数と入力容量、さらに既存のデータでの配線長から得た配線容量から信号遅延時間を見積もるモデルである[2]。しかし、配線形状が見積もれないため、クロストークやヴィア(配線間接続)の抵抗まで考慮することができない。

そこで近年のトップダウン設計では、レイアウト設計を行い、タイミング検証を行うが、タイミング検証で不良となると上位レベルに戻って回路修正が必要となり場合によっては数十回繰り返す。トップダウン設計においてレイアウト設計は自動化が進んでいる段階ではあるが、大量のデータを扱うため最も設計時間を要する処理の1つとなっている。機能設計や論理設計から数十回以上の繰り返し修正することは膨大な時間を要する。

これらの修正の設計期間短縮のために一部修正技術(ECO: engineering change order)が導入されている。これは、タイミング検証で不良となった部分のみを論理設計からレイアウト設計まで一部修正して設計期間を短縮する方法である。しかし、論理回路の修正による回路面積増加や配線結線の変更が、既存のレイアウト設計で修正

可能かどうかについて保証はない。

論理設計において、レイアウト設計を見積もる方法として仮レイアウトによるタイミングの見積もり方法が用いられる。仮レイアウトとは、仮に決めた論理回路を用いて回路素子をだまかに配置し、これらの端子位置からだまかな配線により配線形状や配線遅延を見積もり方法である。しかし、レイアウト設計は論理回路が類似しても結果が大きく変わることや、仮の配置配線そのものが処理時間を要するという課題が残る。

**VLSI 製造が今後さらに微細化することから、設計フローにおいてレイアウト設計結果を早期段階で短時間に精度高く見積もる技術が重要になる[3].**

本研究は、VLSI 設計フローの QTAT(Quick Turn Around Time)化のための超高速レイアウト設計を可能にする超高速配線法について行う。超高速な配線手法が確立すれば、配線詳細まで含めた高精度なタイミング見積もりが短時間に可能となる。ここで TAT は、入力から出力までの一連の処理に要する時間のことで、ここでは VLSI 設計フローにおける仕様設計からタイミング検証までの設計時間を指す。

本研究の超高速配線法は、2 端子ネットを L 型、逆 L 型の単純な配線生成し、他ネットとの配線重複が最少となる配線経路を求める手法に基づいている[4]。本研究では、これを多端子配線に拡張する。多端子ネットは、まず 2 端子へ分割し配線生成を行い、同一ネットの 2 端子配線の多重評価を避ける処理を加える。

本研究ではこの多重評価の回避の高速化を行う。多端子ネットの 2 端子ペアへの分割割り付けは、多端子の迷路配線形状に近い割り付けができる Prim's MST Algorithm を用いる。ランダムに配置した多端子ネットのテスト配線問題で実験したところ、処理速度は迷路配線法に比べて約 25 倍となった。また、実回路 8bit マイコンの配線問題では約 315 倍にもなることが分かった。

本論文の構成は第 2 章で、従来の配線手法について先行研究の L 型経路を用いた超高速配線法の課題について述べる。第 3 章で、本手法の問題の定式化、アルゴリズムを述べ、第 4 章で実験内容、実験結果、考察について述べる。第 5 章で、本研究のまとめと今後の課題を述べる。

## 2. 配線手法

### (1) レイアウト設計と配線設計

レイアウト設計は、VLSI の回路素子(トランジスタ)の位置決めをする配置設計とそれら素子の端子間を結線する配線設計で構成されている。大規模回路での配置、配線では自動配置、配線法が用いられる。

配線設計は、さらに概略配線と詳細配線の 2 段階で構成されている。概略配線では、配線のだまかな経路を決め、詳細配線では、VLSI 製造プロセスで決められた設計ルール(DR: Design Rule)に示される配線間の幅や配線幅を考慮した経路を決める[5]。概略配線のアルゴリズムに関する多くの研究が行われており、例えば BoxRouter

[6]、FastRoute[7]などがある。詳細配線を自動化するアルゴリズムとして迷路配線法(Lee アルゴリズム)がある[8]。

### (2) L 型経路を用いた超高速配線法

先行研究では 2 端子ネットの超高速配線手法で配線の見積もりが行われていた[4]。一般的な VLSI 回路は 1 ネットのもつ端子数には、ばらつきがあるものの、2 端子ネットの割合は全体の約 6~7 割を占めている。そこで、先行研究で 2 端子ネットの超高速配線が行われた。

先行研究では、L 型経路および逆 L 型経路という単純な配線形状が利用されていた。L 型経路、逆 L 型経路について図 2.3(a),(b)を用いて説明する。L 型経路とは、2 端子間において水平方向に進んでから垂直方向に進んで結線する配線形状である(図 2.3 (a))。逆 L 型経路とは、2 端子間において垂直方向に進んでから水平方向に進んで結線する配線形状である(図 2.3 (b))。

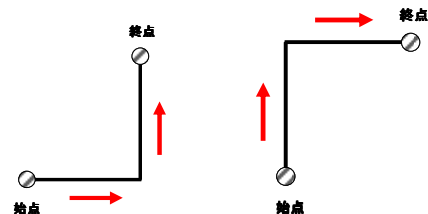


図 2.3(a)L 型経路の例 図 2.3 (b)逆 L 型経路の例

配線領域 80×80 グリッドでネット数 N=30, 50, 80, 100 を実験で評価した結果、迷路配線法に比べ、最大で 20 倍以上の高速化が確認された。配線のショート(配線重複数)は、改善前に比べて 50%以上改善され、また一部の 30 ネットの実験では、完全な詳細配線が得られることも確認できた。

しかし、先行研究の課題として、2 端子間のみ配線手法であるので 3 端子以上の配線に関して配線の見積もりができないため多端子への拡張という課題がある。また、多層が考慮されていないので多層への拡張も必要である。

## 3. 多端子を扱う超高速配線法

### (1) 概要と問題の定式化

多端子ネットの配線 L 型経路および逆 L 型経路の形状で行う。3 端子以上の際は 2 端子ペアに分けて L 型経路、逆 L 型経路を適用する。また配線形状の最適化の際に配線領域の全体を関数で評価し、高速化のため、ネットリストと経路の情報から配線形状を見つける。このとき、ネットリストを x 座標でソートすると、一部ネットにおいて迷路配線法の配線結果の配線形状と大きく違うという問題がある。そこで、迷路配線法の配線結果の配線形状に近づけるために、Prim's MST Algorithm(プリム法)を加えた手法を提案する。

実験の結果より、提案手法では迷路配線法に比べ、実回路で 313.27 倍の高速化が確認された。また配線形状については迷路配線法の配線結果の配線形状に近づいた。

### 問題の定式化

問題: グリッド(格子点) $n \in N(x) \times N(y) = N_g$  における, 配線重なり  $d(n)$  の全グリッド  $N_g$  での総和  $Cost$  を最小化する各ネットの配線形状(L・逆L型経路の組み合わせ)を求める。

問題を評価関数で以下のように定義する。

$$Cost = \sum_{n \in N_g} (d(n) - 1)^2 \quad (1)$$

$Cost$  はグリッド内の現在の重複配線の値である。  $d(n)$  から 1 を引いているのは, グリッド通過数  $d(n)$  の値が  $d(n)=1$  のときは重複する配線がないので考慮しないものとして  $(d(n)-1)$  を計算する。  $(d(n)-1)$  を 2 乗する理由は, グリッド通過数の平均値とばらつきを最適化するためである。

ここで重複配線とは 1 つグリッドにおいて別の配線が 2 つ以上重なっている状態のことを指す。つまり, 1 つのグリッドで  $d$  が 2 以上の値となることである。また, その重なっている配線の数の合計を重複配線数という。実際の配線では, 1 本分のみであるためこれらを除いて評価しなければならない。

重複配線の具体例を以下の図(a)~(d)を用いて説明する。図 3.1(a)~(d)でグリッドを四角で示す。全グリッドの初期値は  $d=0$  とする。まず, 赤線のように配線がされたとする(a)。このとき, 赤線の通過したグリッドの  $d$  は  $d=1$  となる(b)。次に, 緑色の線のように配線されたとする(c)。このとき, 赤色の配線と緑色の配線が一部グリッド内で重なっている。よって, 赤色の配線と緑色の配線の重なっているグリッドの  $d$  は, さらに値 1 がカウントアップして  $d=2$  となる(d)。この  $d=2$  の状態を重複配線という。また, このときの重複配線数は  $6(=2+2+2)$  となる。

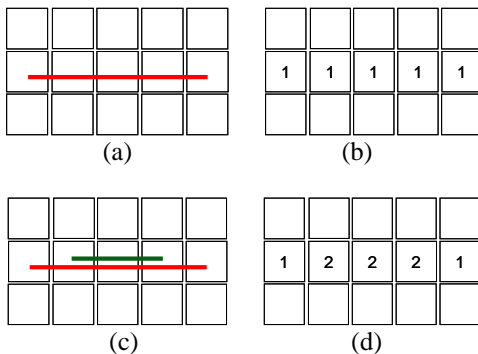


図 3.1 重複配線の具体例

### (2) プリム法のアルゴリズム

配線形状の見積もりをより正確なものにするため, プリム法を使用する。プリム法とは, グラフ理論における重み付き連結グラフの最小全域木問題を解くためのアルゴリズムである。このアルゴリズムの計算量は端子数を  $m$  とし

て  $O(m^2)$  である。図 3.2(a)~(e)を用いてアルゴリズムを説明する。図 3.2(a)のグラフについて最小全域木問題を解く。黄色の円はノード(頂点)を示しており, 円と円をつないでいる線はエッジ(辺)を示しています。また, ノードの中に書かれているアルファベットはノードの名前であり, エッジの近くの数字はそのエッジのコストを示しています。ここで, 図 3.2(a)はエッジには向きが存在しない重み付き連結無向グラフです。

まず, 各エッジのコストを比較し, 最小のコストをもつエッジがつないでいるノードを基準のノードとします。図では, ノード A とノード D をつなぐエッジのコストが 1 で最小になり, 基準のノードにノード A とノード D が選ばれる(図 3.2(b))。次に, 基準のノードから出ているエッジの中で最小のエッジを見つけ, そのエッジがつないでいるノードを基準のノードに加える。なお, この時見つけられるエッジはまだ選ばれていないエッジとし, エッジがつなぐ先は基準のノードに選ばれていないノードにする。図では, 基準のノード D とノード E をつなぐエッジがコスト 2 で最小なり, 基準のノードにノード E が選ばれる(図 3.2(c))。この処理を繰り返すすべてのノードがつながる(図 3.2(d))。この時できるグラフは最小全域木である(図 3.2(e))。

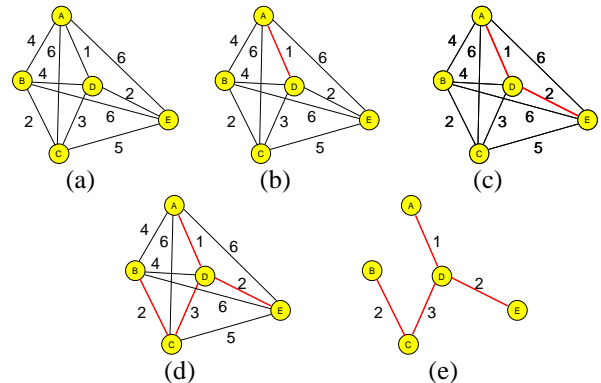


図 3.2 プリム法の説明

### (3) 多端子 L 型配線法

図 3.3 に多端子を扱う超高速配線手法のアルゴリズムの手順を示す。

- Step1. ネットと端子情報を読み込む
  - Step2. 全ネット端子について
    - Step2.1 ネット番号でソートする
    - Step2.2 Prim's MST Algorithmで2端子のペアを見つける
    - Step2.3 2端子ごとに水平, 垂直の直線で配線する(L型)
  - Step3. 重複配線の値(初期コスト)を計算する
  - Step4. 全ネットについて以下の処理を繰り返す
    - Step2.1 配線形状変更前の配線上の重複配線の値(cost1)を計算する
    - Step2.2 L型, 逆L型をランダムで選択し配線形状を変更
    - Step2.3  $cost2 \leq cost1$  ならばこれを受理, そうでなければ元に戻す
  - Step5. 重複配線の値(最終コスト)を計算し, 配線形状を出力する

図 3.3 多端子を扱う超高速配線手法のアルゴリズム

まず Step1 ではネットと端子情報を読み込む。ここでは, 全端子の  $x$  座標,  $y$  座標, ネット番号を読み込む。

次に Step2 では全ネット端子について以下の処理を行う。Step2.1 ではネット番号でソートを行い, Step2.2 では

Prim's MST Algorithm(プリム法)を行い 2 端子のペアを見つける。ここで、この決定した 2 端子のペアごとに取り出し、配線を行う。Step2.3 では初期配線として 2 端子ごとに L 型経路で配線する。ここで、L 型経路とは、2 端子間において水平方向に進んでから垂直方向に進んで結線する配線形状である。

Step3 では初期配線における重複配線の値を計算する。ここでの値を初期コストとする。

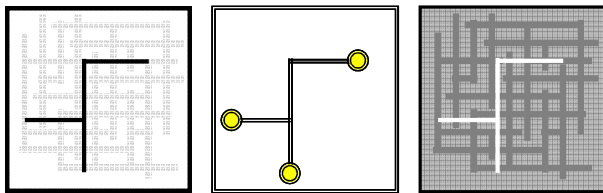
Step4 では全てのネットについて以下の処理を繰り返す。Step4.1 では配線形状変更前の重複配線の値を計算する。この時、重複配線の値を計算する範囲は、各ネットの配線上のみとする。これは、配線領域の全体の重複配線の値を計算するのも、配線上のみの重複配線の値を計算するのも変化する値は同じであると考えたからである。Step4.2 では L 型、逆 L 型経路をランダムで選択し配線形状を変更する。逆 L 型経路とは、2 端子間において垂直方向に進んでから水平方向に進んで結線する配線形状である。変更後、Step4.3 では重複配線の値を計算し  $cost2$  とする。Step4.4 では  $cost2 \leq cost1$ 、つまり配線形状変更によって重複配線の値が削減された場合、この配線形状を保存する。もし  $cost2 > cost1$ 、つまり配線形状変更によって重複配線の値が増加した場合、元の配線形状に戻す。

最後に Step5 では最終コストとして重複配線の値を計算し、配線結果を出力する。

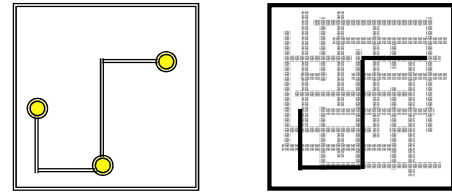
#### (4) 配線変更手順

配線形状の変更手順について図 3.4 を用いて詳しく説明する。

ネットの配線を map へ描くために別の map(map\_bk と呼ぶ)を用いる。map の変更手順は図 3.4 に示す。配線形状の変更前の map の黒色で強調された配線(図 3.4(a))のみを map\_bk に配線密度 1 で書き込む(図 3.4(b))。ここで配線密度とは、単位面積あたりの配線の多さを表わしている。次に、map\_bk に書き込まれた配線形状のみを map から削除し(図 3.4(c))、新たに作った配線形状も同様にペア端子をそれぞれ map\_bk に配線密度 1 で書き込み(図 3.4(d))、map に書き込み(図 3.4(e))、配線の重複量を計算する。このようにすれば、2 端子に分けても配線密度を重複してカウントしないからである。またこのとき、map、map\_bk 上の変更する配線上のみを検索範囲とすることで高速化が見込めると考えられる。



(a)map の様子 (b) map\_bk の様子 (c)map の様子



(d) map\_bk の様子 (e)map の様子  
図 3.4 多端子ネットの配線形状の変更手順

#### (5) 具体例

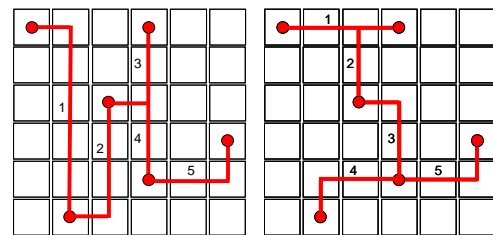
プリム法を利用せず x 座標で端子のソートをおこなった配線法と多端子 L 型配線法の配線形状の違いについて図 3.5(a),(b)を用いて説明する。

図 3.5(a)は x 座標で端子のソートをおこなった配線法での配線形状の模式図を示しており、図 3.5(b)は多端子 L 型配線法での配線形状の模式図を示している。以下の模式図は配線領域 6x6 グリッドで、6 つの端子(赤色の円)で 1 つのネットを構成している。赤色の線は端子と端子を結線する配線を示しており、配線の近くに記している数字は各配線の配線順を示している。また、すべて配線の配線形状を L 型経路でとっている。

x 座標でソートした場合、図 3.5(a)のように端子が選ばれる。したがって、水平方向の配線に比べ垂直方向の配線に偏ると考えられる。

一方で、多端子 L 型配線法ではプリム法を用いて端子ペアを決め、図 3.5(b)のように配線する。したがって、垂直方向の配線の偏りをなくす事ができる、また配線長がより短くなるような経路が選ばれようと考えられる。

図 3.5 では、多端子高速 L 型配線法で総配線長は 18 であるが、最短化多端子高速 L 型配線法で総配線長は 14 となり総配線長が短くなることが確認できる。



総配線長=18 総配線長=14  
(a)x 座標でソート (b)多端子 L 型配線法

図 3.5 配線形状の比較の模式図

## 4. 実験

### (1) 実験

提案手法の速度評価実験と形状評価をおこなった。提案手法を MinGW(C 言語)で実装し、マシン環境は Corei5(2.8GHz)、主記憶 2.96GB、Win7(32bit)で実験をおこなった。

またテスト用回路として、80x80 グリッドにネット数として 30, 50, 80 ネットを乱数より各 20 種類ずつ生成する。実回路として 8bit マイコンの回路を用いた。配線領域は 487x431 グリッドでネット数は 2100 ネットである。



実験の結果より、従来の迷路配線法に比べ、テスト用回路では約 25 倍、実回路では約 380 倍の速度比が確認された。迷路配線法については当研究室で作成した配線プログラム mulrote3 を使用している。詳細は表 4.1 に記しており、各ネット数における端子数は括弧の中に示す。各テスト用回路における処理時間は 20 種類のデータを 100 回繰り返した平均値であり、実回路における処理時間は 8bit マイコンの回路を 100 回繰り返した平均値である。提案手法で 0.08761sec に対して迷路配線法は 27.446sec で速度比は 313.27 倍である。

表 4.1 迷路配線法との速度比較

回路名	データ		処理時間(sec)		速度比
	配線領域	ネット数(端子数)	提案手法	迷路配線手法	
テスト用回路1	80×80	30 (87)	0.00194	0.04635	23.89
テスト用回路2	80×80	50 (156)	0.00259	0.06585	25.42
テスト用回路3	80×80	80 (245)	0.00345	0.0803	23.28
実回路	487×431	2100(6188)	0.08761	27.446	313.27

また、提案手法における初期配線での初期コストの値と、変更後の最終コストを比較した結果は、どのネット数においても平均して 50%以上の改善率が確認されたが、最終コストは 0 にはならなかった。結果については表 4.7(a)~(c)に記す。このときの迷路配線法の最終コストは全て 0 である。各テスト用回路は 20 種類のデータを用いた。表 4.7(c)より、提案手法の初期コストが平均値は 2646.7 で最終コストの平均値は 1239.08 となり 53.2%の改善率が確認された。実回路の配線結果を図 4.3 に示しており、あるネット(クロック)について黄色で強調している。また、図 4.1 には迷路配線法の配線形状を示しており、図 4.2 には多端子 L 型配線法の配線形状を示している。配線形状の一部(白色の円で囲まれている部分)で似た配線形状があることが確認された。

表 4.2(a)30 ネットの提案手法による配線結果

サイズ80×80	提案手法による配線重複		
	30ネット	初期コスト	最終コスト
テスト用回路1-1	381	68.29	82.1
テスト用回路1-2	319	43.82	86.3
テスト用回路1-3	304	44.63	85.3
テスト用回路1-4	341	57.69	83.1
テスト用回路1-5	304	42.80	85.9
テスト用回路1-6	270	71.04	73.7
テスト用回路1-7	303	34.78	88.5
テスト用回路1-8	309	49.59	84.0
テスト用回路1-9	332	49.26	85.2
テスト用回路1-10	280	44.99	83.9
テスト用回路1-11	298	86.90	70.8
テスト用回路1-12	280	40.86	85.4
テスト用回路1-13	330	48.22	85.4
テスト用回路1-14	280	37.97	86.4
テスト用回路1-15	335	58.82	82.4
テスト用回路1-16	382	69.39	81.8
テスト用回路1-17	306	55.56	81.8
テスト用回路1-18	382	56.31	85.3
テスト用回路1-19	306	59.26	80.6
テスト用回路1-20	305	83.81	72.5
平均値	317.35	55.20	82.6

表 4.2 (b) 50 ネット提案手法による配線結果

サイズ80×80	提案手法による配線重複		
	50ネット	初期コスト	最終コスト
テスト用回路2-1	1111	368.50	66.8
テスト用回路2-2	1165	456.55	60.8
テスト用回路2-3	1087	369.20	66.0
テスト用回路2-4	1088	453.74	58.3
テスト用回路2-5	1087	354.10	67.4
テスト用回路2-6	964	445.81	53.8
テスト用回路2-7	1086	350.79	67.7
テスト用回路2-8	1204	391.95	67.4
テスト用回路2-9	1079	412.10	61.8
テスト用回路2-10	1110	397.47	64.2
テスト用回路2-11	969	437.59	54.8
テスト用回路2-12	1110	384.60	65.4
テスト用回路2-13	1067	419.93	60.6
テスト用回路2-14	1098	378.04	65.6
テスト用回路2-15	1205	408.16	66.1
テスト用回路2-16	1071	462.04	56.9
テスト用回路2-17	1112	401.34	63.9
テスト用回路2-18	1071	453.43	57.7
テスト用回路2-19	1119	411.11	63.3
テスト用回路2-20	932	423.86	54.5
平均値	1086.75	409.02	62.4

表 4.3 (c) ネット数 80 の提案手法による配線結果

サイズ80×80	提案手法による配線重複		
	80ネット	初期コスト	最終コスト
テスト用回路3-1	2594	1221.93	52.9
テスト用回路3-2	3062	1269.21	58.5
テスト用回路3-3	2490	1217.31	51.1
テスト用回路3-4	3078	1294.57	57.9
テスト用回路3-5	2415	1225.59	49.3
テスト用回路3-6	2551	1321.92	48.2
テスト用回路3-7	2414	1205.20	50.1
テスト用回路3-8	2638	1275.71	51.6
テスト用回路3-9	2992	1252.27	58.1
テスト用回路3-10	2422	1196.81	50.6
テスト用回路3-11	2550	1319.32	48.3
テスト用回路3-12	2417	1181.29	51.1
テスト用回路3-13	3046	1264.55	58.5
テスト用回路3-14	2495	1171.39	53.1
テスト用回路3-15	2486	1208.97	51.4
テスト用回路3-16	2992	1266.40	57.7
テスト用回路3-17	2372	1165.24	50.9
テスト用回路3-18	2992	1238.78	58.6
テスト用回路3-19	2393	1192.79	50.2
テスト用回路3-20	2535	1292.42	49.0
平均値	2646.7	1239.08	53.2

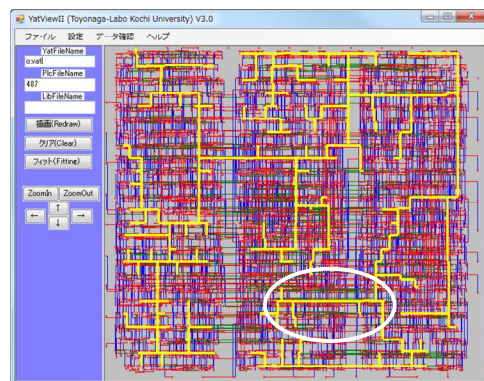


図 4.1 迷路配線法での 8bit マイコンの配線形状

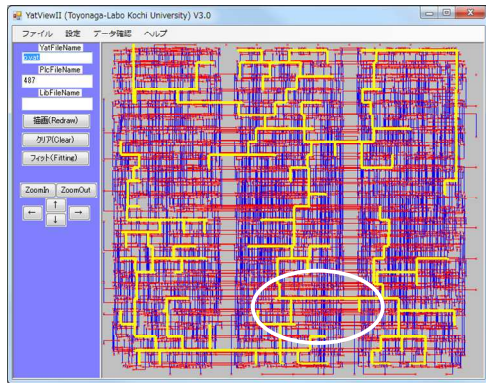


図 4.2 提案手法での 8bit マイコンの配線形状

## 5. まとめ

本論文で提案する超高速配線法は、2 端子ネットを L 型、逆 L 型の単純な配線生成し、他ネットとの配線重複が最少となる配線経路を求める手法に基づいている。本研究では、これを多端子配線まで拡張する。多端子ネットの扱いは、まず 2 端子へ分割し配線生成を行い、同一ネットの 2 端子配線が多重評価されるのを避けるためグリッドマップを使う。多端子ネットを 2 端子ペアへの分割割り付けは、多端子の迷路配線形状に近い割り付けを Prim's MST Algorithm により行う。ランダムに配置した多端子ネットのテスト配線問題で実験したところ、処理速度は迷路配線法に比べて約 25 倍となった。また、実回路 8bit マイコンの配線問題では約 315 倍にもなることが分かった。配線形状は迷路配線と一部似た形状を得ることが確認された。以上のことから、タイミング見積りに寄与できることと考えられる。

今後の課題として、多層を考慮したより精度の高い見積もりができるアルゴリズムへの改善が挙げられる。具体的には、提案手法では配線層を 2 層であるので配線層を多層に拡張することが挙げられるより正確な見積もりができることと考えられる。

## 謝辞

本研究は、科学研究費補助金（JSPS 科研費 22500049）の助成をうけたもので、独立行政法人日本学術振興会に感謝いたします。

## 参考文献

- [1] 小林 正, “多様化するプリント配線板の技術動向”, [http://www.busicompost.jp/tech/tech\\_eptsp201203-3/3/](http://www.busicompost.jp/tech/tech_eptsp201203-3/3/), 2013.
- [2] 細田浩希, 島沢貴美, “Design Wave Magazine 2002 August,” pp.66-67, 2002.8.
- [3] Ahmed F. Abo-ElHadeed, Amr Essam, Amr M.S. Tossou, Ahmed Ramadan, Mohamed Dessouky, Mentor Graphics Corp. “Location Is Everything: Improving Performance with Interactive LDE Estimation,” EETimes, 2012.
- [4] 斉藤未来, 豊永昌彦, “L 型経路を用いた超高速多層配線法,”

Technical Reports on Information and Computer Science from Kochi, 2012.

- [5] Yanheng Zhang, Chris Chu, “GDRouter: Interleaved Global Routing and Detailed Routing for Ultimate Routability,” IEEE, pp. 597-602, 2012.
- [6] K. Yuan M. Cho, K. Lu and D. Z. Pan. Boxrouter 2.0: Architecture and implementation of a hybrid and robust global router. In Proc. Intl. Conf. on Computer-Aided Design, pp. 503–508, 2007.
- [7] Y.Xu, Y.Zhang, and C.Chu. FastRoute 4.0: Global router with efficient via minimization. In Proc. Asia and South Pacific Design Automation Conf., pp. 576–581, 2009.
- [8] C. Y. Lee, “An algorithm for path connections and its application,” IRE Trans. Electron. Comput., pp 346-365, 1961.
- [9] 和田秀三, “確率の基礎”, 株式会社サイエンス社, 1997.2.25.
- [10] 岡村歩, 豊永昌彦, “物理設計予測に向けた多端子を扱う超高速配線法の検討”, 平成 24 年度電気関係学会 四国支部連合大会講演論文集 1-5, 2012.