設 計 自 動 化 66－3
（１９９３．３．１２）

ロングラインに対応した階層的FPGA配線手法

戸川 望　粟島 亨　金子 一哉
佐藤 政生　大附 辰夫

早稲田大学理工学部

〒169　東京都新宿区大久保3-4-1

FPGAは，ローカルライン，ロングライン等のように目的に応じた配線資源を備えているため，これらを有効に利用するような柔軟性に富んだ手法が必要である．本稿では，様々な配線構造に対応し，かつ遅延制御を実現した柔軟な階層的配線手法を提案する．提案手法は，領域を再帰２分割し分割線上のネットの通過位置を２段階の線形割当てによって決定するという高速な処理にもとづいており，さらにロングラインを考慮した割当て処理を行う．このとき，遅延の上界値をネットの特定の端子対に付加し，その値を越えないような配線設計を行うことでクリティカルパスの遅延の最小化を目指す．計算機実験の結果，従来手法に比較して最大30%程度の遅延値の改善を確認した．

A Top-Down Hierarchical FPGA Routing Algorithm Applicable to Long-lines

Nozomu TOGAWA, Toru AWASHIMA, Kazuya KANEKO,
Masao SATO, and Tatsuo OHTSUKI

School of Science and Engineering, Waseda University

3-4-1 Ohkubo, Shinjuku-ku, Tokyo 169, Japan

Field-Programmable Gate Arrays (FPGAs) have been attracting attentions. They posses some kinds of wiring segments such as local-lines and long-lines for connections. Therefore, the wiring segments should be handled as efficiently as possible in wiring design. In this paper, a top-down hierarchical FPGA routing algoritm applicable to long-lines is presented. It is based on fast top-down bi-partitioning and linear assignment. The algorithm assigns nets crossing each bi-partitioning line (cut-line) to wiring segments including long-lines by two-phased linear assignments. It realizes delay control based on maximum delays for connections. Experimental results show its efficiency.

1. まえがき

FPGA（Field-Programmable Gate Arrays）とは比較的高い集積度を持つプログラマブルデバイスの一種であり，とくにシステムのラピッドプロトタイピングの分野で重要なデバイスとなっている．現在までに様々なFPGAが提案されているが[4],[7],[8],[12]，小規模論理を実現する論理ブロックとこれらの接続を行う配線資源によって構成される点で共通している．設計者は単純なスイッチ素子（SRAMにより駆動されるpass-transistorあるいはanti-fuse等）の状態をプログラムすることで，所望の回路を短期間でFPGA上に実現できる．すなわち，FPGAの設計はスイッチ素子の状態を決定することがその目的となる．

FPGAの配線資源に着目すると，一般に配線の接続を決定するスイッチブロックと配線セグメントに分類できる[7],[8]．配線セグメントはさらに設計者の様々な結線要求に対応するため，長さによりローカルライン，ロングラインに分類できる．ローカルラインは短い配線セグメントであり，近い位置にある論理ブロックの接続に適している．一方，ロングラインはローカルラインに比較して長い配線セグメントであり，遠い位置にある論理ブロックを小さな遅延で接続するのに適している．このようにFPGAはいくつかの配線資源を持つため，FPGAの配線設計では，これらの配線資源とくに長さの異なる配線セグメントを効果的に割り当てる設計手法が要求される．

長さの異なる配線セグメントに対応したFPGA配線手法として文献[2]の手法がある．この手法は比較的柔軟性の高い手法であるが，迷路法を基本とした詳細配線手法であるため，前処理として概略配線を必要とし，高速な処理を期待できない．これに対し，我々は高速なトップダウン処理にもとづく階層的概略詳細配線手法を提案した[1]．また，この手法に対しネットに適当な優先度を付加することで配線遅延の制御を実現した[13]．このとき，文献[1],[13]では，ローカルラインのみを処理対象としたが，手法の拡張性は高く，前述のような異なる長さの配線セグメントに対する要求にも応じることが可能である．

そこで本稿では，文献[1],[13]の手法を拡張し，長さの異なる配線セグメントに対応した階層的配線手法を提案する．さらにクリティカルパスを構成するネットの特定の端子対に対して，配線遅延の最大許容値を設定し，その値を越えない範囲で配線処理を行うことで，文献[13]の手法に比較してより洗練された遅延制御を実現する．また，提案手法を計算機上に実装し，評価実験を行った結果について報告する．

2. 問題の定式化

本節ではFPGAモデルおよび配線問題を定義する．

2.1 FPGAモデル

文献[8]で提案されているFPGAアーキテクチャをもとに，図1のようなFPGAレイアウトモデルを定義する．正方形の領域に論理ブロック（logic block）とスイッチブロック（switch block）と呼ばれる2種類のブロックがアレイ状に整列している（ブロックは矩形のシンボルで表す）．各論理ブロックは隣接するスイッチブロックに入出力端子を介して接続される．ローカルラインは隣接するスイッチブロックの間隔と等しい長さを持ち，その間に配置される．一方，ロングラインはいくつかのスイッチブロックをバイパスし，互いに離れたスイッチブロック間に配置される．

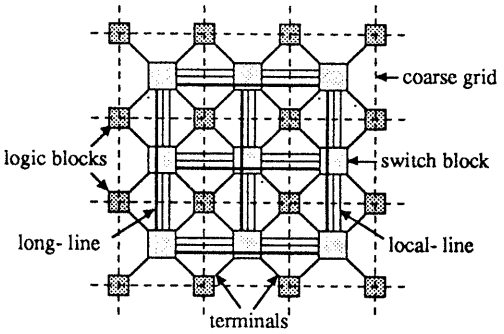

図1 FPGAレイアウトモデル

ここで，論理ブロックおよびスイッチブロックの機能についてまとめる．

論理ブロックは，プログラムにより比較的小規模な論理を実現する．論理ブロックの入出力端子は隣接する4つのスイッチブロックに接続されている．また，周辺部の論理ブロックはパッドに接続され，I/Oブロックの機能を持つ．

スイッチブロックは，一種のスイッチボックスであり，プログラムによって配線セグメントの接続，論理ブロックの入出力端子の接続をスイッチする(図2参照)．

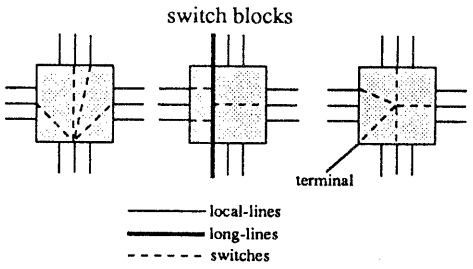

図2 スイッチブロック内部のスイッチパタン

配線セグメントの接続とは，スイッチブロックに接するローカルラインおよびロングラインの接続である．スイッチブロックの1辺に接するローカルラインは，残り3辺に接する配線セグメントと接続可能である．スイッチブロックをバイパスするロングラインは，両側面に位置する配線セグメントと接続可能である．また，スイッチブロックを始点・終点とするロングラインは，ローカルラインと同様に残り3辺に接する配線セグメントと接続可能である．どの配線セグメントと接続可能であるかは，スイッチブロック内部のスイッチパタンによって決

まる．

論理ブロックの入出力端子の接続とは，論理ブロックの入出力端子とスイッチブロックに接する配線セグメントとの接続のことである．論理ブロックの端子は，スイッチブロックに隣接する他の論理ブロックの端子および4辺に接する配線セグメントのすべてあるいは一部に接続することができる．どの端子とどの配線セグメントが接続可能であるかは，スイッチブロック内部のスイッチパタンによって決まる．

2.2 FPGA配線問題

ネットNを結線要求のある論理ブロックの端子集合とする．ネットNはただ1つの信号のソース端子sを含む．ネットNのsを除くすべての端子は信号のシンク端子である．ネットの集合をネットリストとする．

つぎに，ネットNのソース端子からシンク端子に信号が至る遅延時間を定義する．一般的なFPGAの配線遅延を考えたとき，メタルによる配線部分よりもスイッチ素子およびスイッチ素子に付随するバッファ等による遅延がはるかに大きい．これは，FPGAの配線遅延は信号が経由するスイッチ素子数が支配的であることを意味する[2],[7]．そこで，文献[2]と同様にソース端子s∈Nからシンク端子t∈Nに至る遅延時間を，配線径路上で経由するスイッチ素子の数に等しいと定義する．

さらに，各ネットNのシンク端子tには配線の前処理として最大遅延値$d_{max}(s,t)$が与えられるとする．

以上のもとにFPGA配線問題を以下のように定義する．

［定義1］FPGA配線問題とは，

（1）論理ブロックの配置

（2）ネットリストNL

（3）各シンク端子tに対しソース端子sからの最大遅延値$d_{max}(s,t)$

（4）隣接するスイッチブロックを結ぶローカルライン数およびロングライン位置

（5）スイッチブロック内部のスイッチパタン

が与えられたとき，NLに含まれる各ネットNが経由するスイッチブロックおよび配線セグメントを，スイッチブロック内部のスイッチパタンに矛盾なくかつソース端子s∈Nからシンク端子t∈Nに至る遅延時間が$d_{max}(s,t)$を越えないように割り当てることと定義する．

3. ロングラインに対応した階層的FPGA配線手法

本節では，まず用語の定義を行い，続いて提案する配線手法について述べる．

3.1 用語の定義

提案する配線手法は，配線領域に仮想的に設定した粗い格子（coarse grid）を処理の基本単位とする．格子は図1のFPGAモデルに点線で示すように，各格子点が論理ブロックに対応するように設定する．以下，設定した格子に対して用語を定義する（図3参照）．

配線ブロック（routing-block）：格子によって区切られる1区画．図1に示した格子の設定により，各配線ブロックは1つのスイッチブロックを含む．

ブロック境界（section）：配線ブロックの4つの境界線分のうちの1つ．

部分領域（partition）：隣接した配線ブロックの集合から構成される矩形領域．

カットライン（cut-line）：1つの部分領域を2つの部分領域に分割する水平または垂直線分．ブロック境界上に引かれる．

仮端子（pseudo-terminal）：カットラインによって分割された2つの領域の接続を保持するための仮想的な端子．仮端子は，カットラインと交差するロングライン・ローカルライン上に配置される．

容量（capacity）：1つのブロック境界を通過できるネットの最大数．

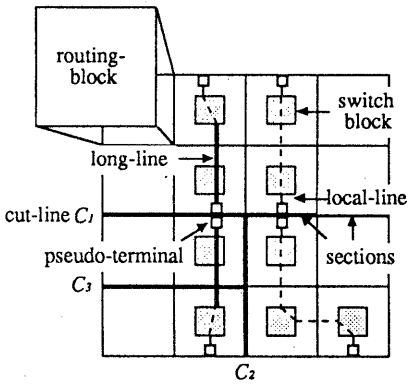

図3 用語の定義

3.2 アルゴリズムの概要

提案する配線手法は，FPGAを対象とした階層的概略詳細配線手法[1],[13]を拡張したものである．この配線手法は，文献[9],[10]の手法をFPGAに対して拡張したものであり，処理対象としてローカルラインのみを考える．処理は配線領域をカットラインによって再帰的に2分割し，カットラインを横切るネットの通過位置を2段階の線形割当てによって決定することを基本とする．この基本処理を再帰的に繰り返すことにより，全ネットの配線径路が決定する．しかし，前述の通りFPGAは，ローカルライン，ロングラインを有する．そこで文献[1],[13]の手法に対しロングラインを導入することを考える．

文献[1],[13]の手法では，カットラインの両側に端子を持つネットに対し，まずカットライン上で通過するブロック境界を決定し（第1段階の線形割当て），その後ブロック境界上に存在するローカルラインを割り当てる（第2段階の線形割当て）．割当ての結果は，仮端子をローカルライン上に配置することで保存される．いま，配線セグメントの割当てを行う第2段階の線形割当てに着目する．このとき，ローカルラインと共にブロック境界と交差するロングラインをも割当て対象の配線セグメントとすれば，文献[1]の手法をロングラインの割当てに対しても適用することが可能である．このとき配線セグメントの長さを評価する必要がある．また，図3のカットラインC_3のように，あるネットにすでに割り当てられたロングラインがカットラインと交差する場合が生じ

る．このようなネットは，C_3上の通過位置が決定していると考えられ，このネットおよび割当て済みのロングラインはC_3における線形割当て処理から除く必要がある．

さらに，文献[13]の手法ではあらかじめネットに与えられた一定の優先度にもとづいて配線遅延の制御を目指していた．ところが，信号のパスを構成するのはネットではなく，論理ブロックおよびその端子対を接続するネットの一部分である．さらに文献[1]の手法が階層的なトップダウン処理を基本とし，処理の進行に従って配線径路の詳細が決定されるという性質に着目すると，各階層で時間的にクリティカルな端子対が，配線遅延を見積もることにより動的に特定でき，これを優先的に遅延の少ない配線径路に割り当てることが可能である．その結果，文献[13]の手法に比較して，クリティカルパスの遅延を小さくすることが期待できる．

これらの割当ては，2段階の線形割当てのコスト設定に依存する．そこで，文献[1],[13]の手法に対し，主にコスト設定の点から改良を加え，ロングラインに対応し，かつより洗練された遅延制御を実現する手法を提案する．

以下に提案手法のアルゴリズムの概要を示す．

[FPGA配線アルゴリズム]

Step1.最大遅延値の初期設定を行う．また配線領域全体を部分領域とし，待ち行列Qに登録．

Step2.Qが空でない場合，部分領域RをQから取り出す．Qが空の場合終了．

Step3.R上にカットラインの位置を決定し，Rを2つの部分領域R_1とR_2に分割する．

Step4.R_1とR_2に仮端子を持ち，かつカットラインと仮端子が割り当てられた配線セグメントが交差しないネットを特定する．

Step5.線形割当てにより，Step5で特定した各ネットをブロック境界に割り当てる（第1段階の線形割当て）．

Step6.コストを増加させない範囲で再割当てを行いカットライン上の混雑度を均一化する．

Step7.各ブロック境界について，線形割当てによって各ネットの仮端子を配線セグメントの上に割り当てる（第2段階の線形割当て）．

Step8.部分領域R_1およびR_2がさらに分割可能である場合，Qに登録する．最大遅延値の更新処理を行いStep2へ．

ここで，カットラインは横切るネットが最も多い位置に引くことにする．混雑度の平均化は文献[5]による．線形割当てのアルゴリズムは文献[3]による．

提案手法のもととなるFPGA配線手法[1]は次の特徴を持つ．

（1）再帰処理に基づくため，アルゴリズムが比較的単純である．

（2）概略配線と詳細配線を一括することで高速な処理が可能である．

（3）配線混雑度の均一化が可能である．

（4）複数のネットを一括処理するため，逐次処理における配線順序の問題を回避できる．

提案する配線手法は（1）～（4）の特徴を継承すると共に，さらに以下の特徴を持つ．

（5）長さの異なる配線セグメント（ローカルライン，ロングライン）に対応する．

（6）文献[13]の手法に比較して，クリティカルパスの遅延をより小さく抑えることができる．

以下では，第1段階（Step5）および第2段階（Step7）の線形割当てにおけるコスト設定，最大遅延値の更新処理（Step8）について述べる．

3.3 第1段階の線形割当て

カットラインの両側に端子を持ち，かつカットラインと仮端子が割り当てられた配線セグメントが交差しないネットに対し適当なブロック境界1つを線形割当てによって割り当てる．割当ての対象ネット集合をN，カットライン上のブロック境界の集合をSとする．また，ブロック境界jと交差する配線セグメントの中でネットが割り当てられていない配線セグメントの数（容量）をc[j]，ネットiをブロック境界jに割当てたときのコストをa[i,j]とする．またネットiが境界jに割り当てられていたら1，そうでなければ0をとるような0-1変数をx[i,j]と表すことにすると，第1段階の線形割当ては以下のように定式化できる．

$$\text{minimize} \sum_{i\in N}\sum_{j\in S} a[i,j]\cdot x[i,j]$$

$$\text{subject to} \sum_{i\in N} x[i,j] \leqq c[j] \ (\text{for each } j\in S)$$

文献[13]にあるように，FPGAの配線設計では遅延制御を実現する必要がある．このため，ここでは割当ての対象ネットiのシンク端子tにソース端子sからの最大遅延値$d_{max}(s,t)$が与えられたときに，遅延時間をその範囲内に抑えることを目指したコスト設定を提案する．

いま，カットラインによって分割された2つの部分領域をR_1とR_2とし，ネットiはR_2にソース端子sを持つとする．そしてiのブロック境界jへの割当てを考える．

まず，2端子s，t間の遅延時間の見積りを行う．ロングラインは，遠距離に位置する端子間を少ないスイッチ素子数で結線可能なため，遅延時間を小さくすることが可能である．したがって，ロングラインを数多く用いて端子s，t間を結線すれば，遅延時間を小さくすることが可能である．しかし，一般にローカルライン数に比較してロングラインの数は少なく[7],[8]，最終的にロングラインに割り当てられるネットは少数に限られる．そこで，すでにそのネットに割り当てられたロングラインを利用した上で，未配線部分をローカルラインのみにより最短径路で配線した場合の遅延時間を遅延見積値として採用し，つぎのように算出する．

端子（仮端子）s，t∈iの各々を含む2つの配線ブロックを頂点とする最小の矩形領域B(s,t)を考える（図4）．B(s,t)において縦横の配線ブロック数をBX，BYとすると，(BX+BY-1)は2点間をローカルラインのみで結線したときの遅延時間を表している．また，ネットiが多端子間ネットで，iに割り当てられた仮端子p（ロングライン上

に配置されている）がB(s,t)内部でBL個のスイッチブロックをバイパスすれば，BL個だけ経由するスイッチ素子数が少なくなり，遅延時間も減少する．そこで，s，t間の遅延見積値$d_{est}(s,t)$を

$$d_{est}(s,t) = (BX + BY - 1) - BL$$

と定義する．図4ではBX=4，BX=3，BL=2であるため，$d_{est}(s,t)=4$となり，点線のようにローカルラインで結線したときのs，t間の遅延値に等しい．また，同様にブロック境界jと端子tに対しても$d_{est}(j,t)$を定義できる．

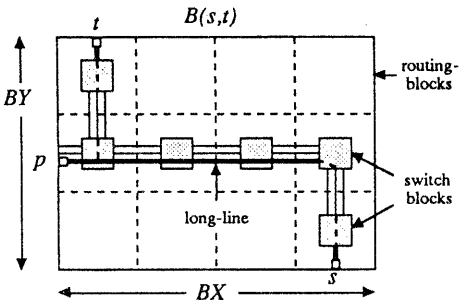

図4　配線遅延の見積値

R_1に含まれるネットiのシンク端子tに対して，$slack(s,t)=d_{max}(s,t)-d_{est}(s,t)$とする．slack(s,t)の値は，処理階層によって動的に更新され，この値が小さいほど，そのソース・シンク端子対はクリティカルであるといえる．このような端子対は遅延の小さい径路で結線するのが望ましい．また，端子sからブロック境界jを通り，端子tに至る径路の遅延見積値$d_{est}(s,j,t)$を$d_{est}(s,j)+d_{est}(j,t)$で与える．$d_{est}(s,j,t)$と$d_{est}(s,t)$との差が小さいブロック境界jほど，迂回が小さく，ゆえに遅延も小さい径路を与えるといえる．さらに，$d_{est}(s,j,t)$が$d_{max}(s,t)$を越えるブロック境界jは，最大遅延値内で配線することが難しいと考えられるため，割当てを抑制するためコスト∞を与える．以上の考察のもとに，シンク端子tに対し，ブロック境界jのコストa'[t,j]を以下のように定義する．

$$a'[t,j] = [d_{est}(s,j,t) - d_{est}(s,t)] / slack(s,t) \quad (\text{if } d_{max}(s,t) \geqq d_{est}(s,j,t))$$

$$a'[t,j] = \infty \quad (\text{ohterwise})$$

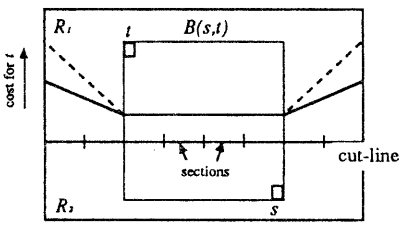

図5　第1段階の線形割当てコスト

図5に単純な2端子間ネット（端子s，tの結線）の場合のコストの例を示す．上式により，矩形B(s,t)に含まれるカットライン上のブロック境界は遅延が最小の径路を与えるため最小コストになり，遠ざかるに従いコストは増加する．コストの増分はslack(s,t)が小さいほど大きくなる．すなわち，クリティカルな端子対ほど遅延の小さな径路に割り当てられることが期待される．

以上の和をとって，ネットiをブロック境界jに割り当てるコストa[i,j]を以下のように定義する．

$$a[i,j] = \sum_{t \in i} a'[t,j]$$

上式において，和は部分領域R_1に含まれるネットiのすべての端子（仮端子）を意味する．

ネットのソース端子がR_1に含まれる場合も同様である．

3.4　第2段階の線形割当て

ブロック境界を割り当てられたネットを適当な配線セグメント1つに線形割当てによって割り当てる．いま，1つのブロック境界に割り当てられたネット集合をM，ブロック境界と交差する割当て可能な配線セグメントの集合をW，ネットiを配線セグメントjに割り当てたときのコストをb[i,j]とする．またネットiがトラックjに割り当てられていたら1，そうでなければ0をとるような0-1変数をy[i,j]と表すことにすると，第2段階の線形割当ては以下のように定式化できる．

$$\text{minimize} \sum_{i \in M} \sum_{j \in W} b[i,j] \cdot y[i,j]$$

$$\text{subject to} \sum_{i \in M} y[i,j] \leqq 1 \ (\text{for each } j \in W)$$

文献[1]における第2段階の線形割当てでは，ローカルラインのみを対象としているため，（1）スイッチブロック内部のスイッチパタンに矛盾しない割当てを考えれば十分であった．しかしながら，ロングラインを考えた場合，配線セグメントの長さに関するコストが必要である．たとえば，カットライン位置に近い端子が，ロングラインに割り当てられた場合，ロングラインの大部分が無駄な配線となる．また，ロングラインはネットの遅延を減少させることが可能である．そこで，ここでは（1）に加えて（2）無駄な配線セグメントを生じない割当て，（3）クリティカルな端子対に対し優先的なロングラインの割当て，を考え，各々を達成するコスト設定を提案する．コストは文献[1]と同様にカットラインの両側の部分領域について別個に算出し，これらを加え合わせることにする．以下，1つの部分領域R_1に着目し，カットライン上の1つのブロック境界Sでの配線セグメントの割当てを考える．また，ネットiをSに割り当てられたネットの1つとし，iをSと交差する配線セグメントjに割り当てる場合のコストについて考える．

（1）スイッチパタンに関するコスト：スイッチブロック内部のスイッチパタンに矛盾しない割当てを達成するためには，文献[2]のように配線資源の全探索を行い，スイッチパタンに矛盾しない配線径路を発見するのが最も簡単である．しかしながら，文献[1]は配線資源の全探索は必ずしも必要なく，ブロック境界近傍の局所領域（local region）における径路探索で十分であることを実験的に示している．そこで，この局所領域の概念をロングラインに適用するように拡張し，ここでも同様な局所

領域内での径路探索によるコストを考える．

ロングラインは，ローカルラインよりも長い配線セグメントであるため，その長さに相当する分だけ局所領域を大きくとる必要がある．そこで，１つの配線セグメント（ロングライン）j_1に対し，図６のようにj_1を覆う最小の配線ブロック集合（黒線の中）およびこれらに隣接する配線ブロックからなる領域を新たに配線セグメントj_1の局所領域と呼ぶ．つぎに，R_1に含まれるネットiの端子（仮端子）tを考える．tが局所領域の内部に存在する場合，tから配線セグメントjに至る径路のうち，最小のスイッチブロックを経由する径路を可能径路といい，その数pr(t,j)を可能径路数と呼ぶ．可能径路数は，経由するスイッチブロックのスイッチパタンにより算出することができる．一方，tが局所領域の外部に存在する場合にはjに至る可能径路が数多く存在すると考えられる．そこでこの場合，径路探索は行わず，局所領域内部のどの端子に対する可能径路数以上の値として，１ブロック境界上に存在する配線セグメントの総数を可能径路数とする．

可能径路の例として，図６の局所領域の場合を考える．ネットiの仮端子t_1は局所領域の内部に存在し，仮端子t_1からロングラインj_1に至る可能径路数は$pr(t_1,j_1)=2$と求めることができる．同様に$pr(t_2,j_1)=3$となる．

以上のように定義した可能径路数をもとに（１）に対するコスト$b_1[i,j]$を以下のように定義する．

$$b_1[i,j] = 1 / \sum_{t \in i} pr(t,j)$$

上式のコスト設定方法によって，配線セグメントに至る可能径路が多いネットほど小さなコストが与えられる．これは，配線セグメントに至るまでの可能径路数が多いほど，領域の分割が進んだとき，配線セグメントの割当てが行いやすくなると考えたためである．

さらに，スイッチパタンに矛盾なく割当てを行うため，文献[1]と同様に，１つのネットiに属する端子（仮端子）tが配線セグメントjを覆う最小の配線ブロック集合の中に存在し，jとtとを接続するスイッチが存在しないときは，$b_1[i,j]=\infty$とする．

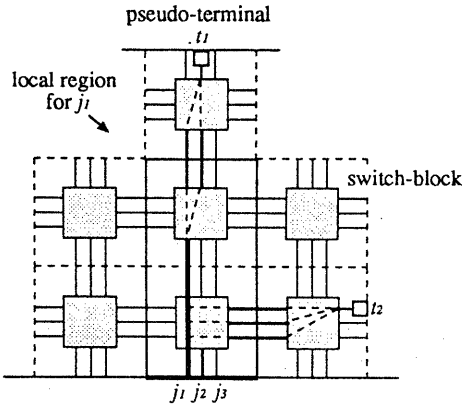

図６　第２段階の線形割当て

（２）配線セグメントの長さに関するコスト：図６において仮端子t_2を考える．t_2がロングラインj_1に割り当てられた場合，j_1の上側１配線ブロック分は無駄な配線となる．配線セグメントの限られるFPGAではこのような無駄な配線を小さくする必要がある．

そこで，まず端子（仮端子）tからカットラインまでの距離をD(t)，配線セグメントjの長さをL(j)とする．D(t)はtが含まれる配線ブロックとカットラインとの最小距離を配線ブロック単位で表したものであり，L(j)はjを覆う最小の配線ブロック集合の数で定義する．図６の例では，$D(t_1)=3$，$D(t_2)=1$，$L(j_1)=2$である．このとき，$D(t)-L(j) \geq 0$ならば，カットラインに対し端子tが配線セグメントjより離れて位置する．これは，端子tがロングラインjに割り当てられても無駄な配線が生じないことを意味する．したがってこのときのコストを0とする．また，$D(t)-L(j)<0$ならばD(t)-L(j)の値が小さいほど，無駄な配線を生じることになる．そこで，（２）に対するコスト$b_2[i,j]$を以下のように定義する．

$$b_2[i,j] = \sum_{t \in i} f(L(j)-D(t))$$

ただし，f(x)=x（if x>0），f(x)=0（otherwise）とする．上式のコスト設定により配線セグメントよりもカットラインに近い距離にある端子に大きなコストが与えられる．ゆえに，そのような端子の割当ては抑制される．

（３）遅延に関するコスト：図６において仮端子t_1を考える．t_1がロングラインj_1に割り当てられた場合，t_1はカットラインまで１つのスイッチブロックをバイパスする．t_1がローカルラインj_2に割り当てられた場合，バイパスするスイッチブロックは存在しない．

いま，t_1がクリティカルな端子であったとする．すなわち，$slack(s,t_1)$が小さい値を持つ端子であったとする．このとき，ロングラインj_1を割り当てれば，スイッチブロックをバイパスすることが可能であり，配線遅延を減少できる．その結果，遅延に対して余裕を持った配線設計を実現し，最終的に配線遅延を$d_{max}(t_1)$に抑えることが期待できる．また，ロングラインが長いほど配線遅延の減少は大きいことが期待できる．そこで，（３）に対するコスト$b_3[i,j]$を以下のように定義する．

$$b_3[i,j] = \sum_{t \in i} slack(s,t) / L(j)$$

上式によって，同じ端子tに対して配線セグメントjが長いほどコストは小さくなる．また，同じ長さの配線セグメントに対しては，端子tがクリティカルであるほどコストは小さくなる．

カットラインの片側の部分領域R_1におけるネットiのコストb[i,j]は上述の（１）～（３）を考える必要がある．ここでは単純にb_1からb_3を加えて算出する．

$$b[i,j] = b_1[i,j] + b_2[i,j] + b_3[i,j]$$

上記の方法で，ブロック境界の両側において別々に計算された２つのコスト値を加えて第２段階の線形割当てコストとする．また，線形割当てにおいて，コスト∞の配線セグメントに割り当てられたネットは，スイッチパタンに矛盾するため未結線となる．

3.5　最大遅延値の更新処理

カットラインの両側に端子を持つネットは，接続を保つため，上述の処理によって特定されたカットライン上の配線セグメントに仮端子を配置する．

いま，ネットiの仮端子pが配線セグメントjに配置されたとする．簡単のためiが2端子間ネットであり，カットラインによって分割された2つの部分領域R_1，R_2に対しソース端子sをR_2，シンク端子tをR_1に持つ場合を考える．このとき，R_1にはソース端子がないため，以後の処理では新たにpをR_1におけるiのソース端子とする．また，R_2にはシンク端子として新たにpが加わることとなる．

ネットの分割に伴い，最大遅延値$d_{max}(s,t)$を2つに分割する必要がある．いま，

$$slack'(t) = d_{max}(s,t) - d_{est}(s,p,t)$$

とすれば，slack'(t)は余裕時間を表しているといえる．そこで，本手法では，単純に余裕時間を2つの部分領域に均等に分けることで$d_{max}(s,t)$を分割する．すなわち，sからpに至る最大遅延値を$[d_{est}(s,p)+slack'(t)/2]$，pからtに至る最大遅延値を$[d_{est}(p,t)+slack'(t)/2]$とし以後の処理を続ける．多端子間ネットの場合も同様の更新処理を行う．

以上の更新処理により，以後，分割された部分領域に対し処理を再帰的に実行できる．

4．計算機実験による手法の評価

提案した配線手法をSUN SPARC station2(28.5MIPS)上にC言語で実装し，計算機実験によって評価した結果について報告する．データはMCNC組み合わせ回路を用いた．実験は(1)ロングラインを用いず，最大遅延値にもとづくコスト設定のみによる遅延制御の効果，(2)(1)に加えてロングラインを用いたときの遅延制御の効果という2点に着目して行った．また，比較対象として文献[1]によるFPGA配線手法を用いた．以下，実験手順および実験結果について述べる．

4.1 実験手順

以下の方法で実験を行う．

（1）テクノロジーマッピング：入力論理回路を論理ブロックの仕様にもとづき適当に分割する．ここで4入力1出力の任意の組み合わせ論理を実現できる論理ブロックを仮定する[7],[8]．mis-pga[11]と呼ばれるテクノロジーマッパーを用いる．

（2）配置：ネットリストに従い，論理ブロックを配置する．シミュレーティッドアニーリングによるペア交換法を用いている．目的関数は優先度を重みとして乗じた総配線長の最小化である[13]．

（3）配線：提案手法および文献[13]の手法を用いる．

配線処理では，配線セグメントおよびスイッチパタンの情報が必要となる．本実験では，文献[1]と同一のスイッチパタンを用いる．また，隣接するスイッチブロック間に配置するローカルライン数は表2の通りとする．さらに提案手法では，ローカルラインに加えてチャネルを貫通するロングラインを1チャネルあたり2つ配置する．これらの値はすべて実際のFPGAアーキテクチャにもとづいたものである[7],[8]．

また，提案手法では最大遅延値を各端子対に設定する必要がある．そのため，まず，配置後の論理ブロックに対して，すべてのソース・シンク端子対の仮想配線遅延時間を，ローカルラインにより最短径路で配線したと仮定した場合の遅延時間で与える．仮想配線遅延をもとに簡単な遅延解析[6]を行い，各論理ブロックにおける到着時間と要求到着時間の差を評価する．そして，その差が0となる2つの論理ブロックを結ぶ端子対の集合をクリティカルパスとする．最大遅延値の初期値は，クリティカルパスを構成する端子対に対して，仮想配線遅延時間により与えることにする．また，その他の端子対については，最大遅延値∞を与える．

表1　ベンチマーク回路

回路名	論理ブロック数	ネット数	端子数
con1	15	13	35
rd53-hdl	24	21	55
rd53	42	23	87
misex1	42	35	133
z4ml	53	49	197
f51m	87	79	319
rd73	102	99	421
misex2	126	108	361

表2　遅延値の比較

回路名	ローカルライン数	文献[13]			提案手法					
					ローカルラインのみ			ロングラインあり		
		遅延値	総配線長	CPU time[s]	遅延値	総配線長	CPU time[s]	遅延値	総配線長	CPU time[s]
con1	3	21	28	0.09	21	28	0.09	18	30	0.12
rd53-hdl	3	39	56	0.25	39	56	0.12	35	60	0.17
rd53	4	35	89	0.21	34	93	0.28	32	119	0.30
misex1	5	31	154	0.43	29	149	0.38	28	175	0.54
z4ml	6	55	267	0.77	49	250	0.79	47	318	0.80
f51m	6	56	409	1.07	53	435	1.38	51	487	1.32
rd73	8	76	554	1.48	60	545	1.87	58	616	1.84
misex2	6	47	484	1.34	37	507	1.58	36	594	1.77

*表のローカルライン数とは隣接するスイッチブロックを結ぶローカルライン数

*遅延値とは主入力か主出力に至る遅延時間の最大値

4.2 実験結果

表1にテクノロジーマッピング後の回路の諸元を示す．表2に文献[13]の手法と提案手法で遅延制御を試みた場合の遅延値，総配線長および処理時間を示す．ここに，遅延値とはスイッチ素子による遅延を1，論理ブロックによる遅延を3としたときに，信号が主入力から主出力に至る遅延時間の最大値である．また，総配線長は，ローカルラインの配線長を1，ロングラインの配線長を（バイパスしたスイッチブロック数+1）として算出した．また，図7に回路misex2に対して本手法を適用した結果を示す．

表2から，ロングラインを用いない場合，本手法は総配線長を数%の増加に抑えながら，最大20%程度の遅延値の改善を達成している．さらに，ロングラインを用いた場合，最大30%程度の遅延値の改善を達成している．また，このとき従来手法に対する処理時間の増加も同程度に保っている．以上から提案手法がとくに遅延制御の点で従来手法に比較して優位であることが結論づけられる．

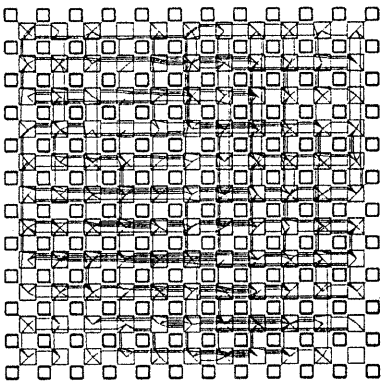

図7　回路misex2の本手法による出力結果

5. むすび

FPGAは，ローカルライン，ロングライン等のように目的に応じた配線資源を備えている．したがって，これらの配線資源を有効に利用するような柔軟性に富んだ手法が必要である．本稿では，従来の再帰2分割と2段階の線形割当てによる階層的FPGA配線処理手法を拡張し，長さの異なる配線セグメントに対応する配線手法を提案した．さらに，ネットの特定の端子対に最大遅延値を設定することでクリティカルパスにおける遅延の減少を実現した．計算機実験の結果，文献[13]の手法に比較して最大30%程度の遅延値の改善を確認した．

現在，FPGAの設計処理をテクノロジーマッピング，配置，配線の3段階に分割して考えているが，これらを有機的に結合し大局的な最適化を行うことが今後の課題である．

謝辞　本研究を進めるにあたり，数々の有益なご助言ご討論を頂きました，川名啓一氏をはじめとする川崎製鐵（株）の関係者の方々に感謝いたします．なお，本研究は文部省科学研究費補助金：奨励研究(A)04855059（平成4年度）「計算幾何学のLSI設計への応用に関する研究」の援助のもとに行われたものである．

文　献

[1]粟島亨，戸川望，金子一哉，佐藤政生，大附辰夫："FPGAを対象としたトップダウン配線手法の実装と評価"，信学技法，VLD92-38(1992).

[2]Brown S., Rose J. and Vranesic Z.: "A Detailed Router for Field-Programmable Gate Arrays", *IEEE Trans. Comput. - Aided Des. Integrated Circuits & Syst*, CAD-11, 5, pp.620-627(1992].

[3]Burkard R.E. and Derigs U.: *Assignment and Matching Problems: Solution Methods with Fortran Programs*, Springer-Verlag(1980].

[4]Gamal A.El, Greene J., Reyneri J., Rogoyski E., El-Ayat K.A. and Mohsen A.: "An Architecture for Electrically Configurable Gate Arrays", *IEEE J. Solid-State Circuits*, SC-24, 2, pp.394-398(1989].

[5]長谷川晃司，粟島亨，佐藤政生，大附辰夫："線形割り当てに基づいた概略配線手法の実装と評価"，信学技法，VLD90-97(1991).

[6]Hitchcock R.B. Sr., Smith G.L. and Cheng D.D.: "Timing Analysis of Computer Hardware", *IBM J. Res. & Dev.*, 26, 1, pp.100-105(1982).

[7]Hsieh H.-C., Carter W.S., Ja J., Cheung E., Schreifels S., Erickson C., Freidin P. and Tinkey L.: "Third-Generation Architecture Boosts Speed and Density of Field-Programmbale Gate Arrays", *Proc. IEEE 1990 Custom Integrated Circuits Conf.*, pp.31.2.1-31.2.7(1990).

[8]Kawana K., Keida H., Sakamoto M, Shibata K. and Moriyama I.: "An Efficient Logic Block Interconnect Architecuture for User-Reprogrammable Gate Array", *Proc. IEEE 1990 Custom Integrated Cirucuits Conf.*, pp.31.3.1-31.3.4(1990).

[9]Lauther U.P.: "Top Down Hierarchical Global Routing for Channelless Gate Arrays Based on Linear Assignment", *Proc. VLSI 87*, pp.109-120 (1987).

[10]Marek-Sadowska M.: "Route Planner for Custom Chip Design", *Proc. IEEE 1986 Int. Conf. on CAD*, pp.246-249(1986).

[11]Murgai R., Nishizaki Y., Shenoy N., Brayton R.K. and Sangiovanni-Vincentelli A.: "Logic Synthesis for Programmable Gate Arrays", *Proc. 27th DA Conf.*, pp.620-625(1990).

[12]Muroga H., Murata H., Saeki Y., Hibi T., Ohashi Y., Noguchi T. and Nishimura T.: "A large Scale FPGA with 10K Core Cells with CMOS 0.8um 3-Layered Metal Process", *Proc. IEEE 1991 Custom Integrated Circuits Conf.*, pp.6.4.1-6.4.4(1991).

[13]戸川望，粟島亨，金子一哉，大附辰夫："遅延を考慮したFPGAレイアウトシステム"，信学'92秋大，SA3-2(1992).