パス長制約を考慮した FPGA 配置概略配線同時処理手法

戸川 望† 佐藤政生† 大附辰夫†

FPGA のレイアウト設計は，一般的な LSI の設計方式にならい配置設計，配線設計という段階的な処理をとっている．ところが，FPGA は規則的，対称的な構造を持つデバイスであり，この構造を最大限に活用すれば，レイアウト設計を一元化することが考えられる．一方，FPGA のプログラムは簡単なスイッチ素子によって実現されるが，スイッチ素子の信号遅延は一般に大きく，ゲートアレイと比較して FPGA の動作速度は遅くなる傾向にある．つまり，FPGA の設計では配線遅延がこれまで以上に重要な問題となり，信号パスの長さ（パス長）の制御を可能としたレイアウト設計手法が要求される．本論文では，FPGA を対象にパス長の制御を可能とした配置および概略配線の同時処理手法を提案する．提案手法は単純な領域の分割処理とそれにともなうブロック集合の分割処理を基本としている．このとき，FPGA の構造の規則性を利用し，ネットの概略径路を仮想的なブロックの並びによって表している．このブロックは，配置すべき通常の論理ブロックと同等であり，その結果，配置概略配線の同時処理が可能となる．また，タイミングがクリティカルな信号パスに対しパス長の上限値（パス長制約）を与える．領域の分割においてパス長の評価値を算出し，その評価値をもとに制約を満たすようにブロック集合の分割を行う．最後に本手法を計算機上に実装した結果を報告し，有効性を示す．

A Timing-Driven Simultaneous Placement and Global Routing Algorithm for Field-Programmable Gate Arrays

Nozomu Togawa,† Masao Sato† and Tatsuo Ohtsuki†

In the design of FPGAs whose programmability is simply realized by means of switches, the routing delay problem is essential and the signal path length should be controlled during layout design. The conventional layout design of FPGAs is composed of two phases, i.e., placement and routing. However, those phases can be treated simultaneously by fully exploiting FPGAs' regular structure. In this paper, an FPGA layout algorithm is presented, which deals with placement and global routing simultaneously. It is based on simple and fast top-down hierarchical bipartitioning, and global routes can be represented by a sequence of pseudo blocks. Since pseudo blocks and logic blocks are processed in a similar way, the placement and global routing are treated simultaneously. For each critical signal path, its upper bound in length is given as a constraint. During the bipartitioning, the algorithm calculates estimated lengths of the paths and, based on them, partitions the set of blocks so as to satisfy the constraints. Experimental results for several benchmark circuits demonstrate its efficiency and effectiveness.

1. まえがき

FPGA (Field-Programmable Gate Arrays) は，比較的集積度の高いプログラマブルデバイスの一種であり[7)~9),13)]，そのレイアウト設計では，通常，問題を簡単化するため，配置設計，配線設計という段階的な処理を行っている[3)]．ところが，FPGA は規則的，対称的な構造を持つデバイスであり，これらを十分に利用することができれば，本来密接に関係のある配置設計と配線設計との統合化が可能であると考える．

一方，FPGA のプログラミングは pass-transistor や anti-fuse といった単純なスイッチ素子によって行われる．スイッチ素子の信号遅延は大きく，その結果，通常のゲートアレイと比較して FPGA の動作速度は遅くなる傾向にある[3)]．したがって FPGA を高速に動作させるためには，信号遅延特に配線遅延を考えた配置配線設計が必須となる．

本論文で提案する手法は，FPGA を対象に配置と概略配線を同時に処理する手法であり，加えて信号パスの長さ（パス長）を指定された上限値以内に抑える工夫をしている．提案手法は次に示すような多くの特

† 早稲田大学理工学部電子通信学科
School of Science and Engineering, Waseda University

長を持つ.

(1) FPGA の規則的な構造を利用し，概略配線の径路を仮想的に設定したブロック（仮想ブロック）の並びとして表している．仮想ブロックは配置すべき通常のブロックと同等に扱うことが可能なため，配置と概略配線の同時処理が可能である．これにより良質な結果を得ることが期待できる．

これまで提案されてきたビルディングブロック等を対象とした配置概略配線の同時処理手法[4),5),14),15)]が単に配置と概略配線を交互に繰り返していただけなのに対し，配置設計と概略配線設計とを完全に同時処理している．

(2) タイミングがクリティカルな信号パスに対し，従来，信号パスをネットに分割しネットの長さに関する制約を課していたが[6),16)]，提案手法では，パス長を直接評価することが可能であり，その結果，効果的なパス長の制御を行うことができる．

(3) アルゴリズムが単純な領域の2分割による再帰処理にもとづくため，簡単かつ高速である．

(4) 機能的に等価な端子を複数持つブロックに対して，ごく自然な方法で端子の割当てを行うことができる．

(5) 多端子間ネットを2端子間ネットに分割することなく，直接扱うことが可能である．

以下，本論文で扱う問題を定式化し提案手法について述べる．

2. 問題の定式化

2.1 FPGA レイアウトモデル

文献 9) によって提案されている 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考える．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ており，各スイッチブロックは水平・垂直のトラックによって互いに接続されている．隣接するスイッチブロック間を接続するトラックの集合をチャネルと呼ぶ．

論理ブロックのうち，最も外側にあるものは，I/O パッドに接続されており，I/O ブロックとしての機能を持つ．I/O ブロックは，隣接するすべてのスイッチブロックに対して入出力端子が存在する．その他の論理ブロックは，プログラムによって4入力以下の小規模な論理（組み合わせ論理等）を実現可能である．入力端子は隣接する4方向のスイッチブロックそれぞれに対して一つずつ存在し，出力端子は左下に隣接するスイッチブロックに対してのみ存在するものとする（**図 2**）．

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論理ブロックの入出力端子および水平・垂直トラックを接続する．

2.2 パス長制約

前述のように，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．信号パスは，スイッチ素子により接続されるトラックによって構成される．したがって，信号が経由するスイッチ素子数は，パスを構成するトラックの数によって決まるということができる．FPGA の場合，パスを構成するトラック数は経由するチャネルを決める概略配線によって特定される．つまり，配置概略配線の同時処理において，タイミングがクリティカルな信号パスを構成するトラック数が指定された上限値以下であるように処理を行えば配線遅延を小さく抑えることが可能である．

以下，パスを構成するトラック数をパス長と呼び

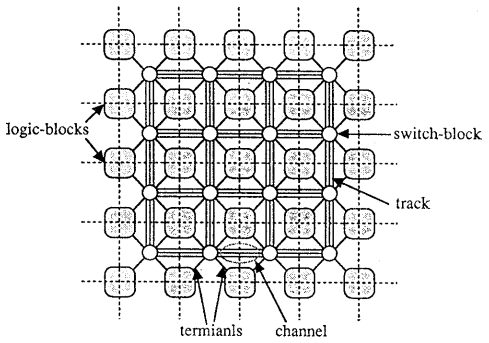


図 1 FPGA レイアウトモデル
Fig. 1 FPGA layout model.

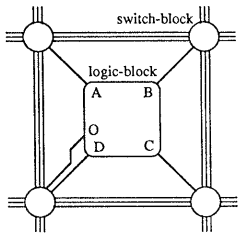


図 2 論理ブロックの端子位置
A, B, C, D: 入力端子，O: 出力端子
Fig. 2 Terminal positions of a logic-block.
A, B, C, D: input terminals, O: output terminal.

(図 3)，パス長を指定された上限値以下に抑えることをパス長制約 (path length constraint) と呼ぶ．

2.3 パス長制約を考慮した FPGA 配置概略配線問題

論理ブロックが配置されるべき場所をスロットと呼ぶ．各論理ブロックの入出力と区別するため，与えられた回路に対する入力および出力を，特に主入力および主出力と呼ぶことにする．配置すべき論理ブロック集合の中で主入力および主出力を与えるものをI/Oブロックと呼ぶ．論理ブロック間の結線要求をネットとし，ネットの集合をネットリストと呼ぶ．

パス長制約は，信号パス p およびそのパス長の上限値 $L_{max}(p)$ が与えられたとき，配置概略配線後の p のパス長 $Lr(p)$ に対し，

$$Lr(p) \leqq L_{max}(p)$$

を満たすように配置概略配線することである．

定義 1 FPGA の配置概略配線問題とは，

(1) 論理ブロック集合およびネットリスト

(2) スロット集合および論理ブロックの端子位置

(3) チャネルを構成するトラック数

(4) パス長制約の集合

が与えられたとき，

(a) 論理ブロックが配置されるスロット位置

(b) ネットの端子位置

(c) ネットが経由するチャネル

をチャネルを構成するトラック数を越えることなく，かつ，パス長制約を満たすように割り当てることである．このとき，I/Oブロックは最も外側のスロットに配置することとする．

図4にネットリスト，パス長制約とこれらに対するレイアウト結果の例を示す．

3. パス長制約を考慮した FPGA 配置概略配線同時処理手法

3.1 用語の定義

アルゴリズムの提案に先だって，用語を定義する．

単位格子：図1に示す格子（点線）によって区切られた1区画．

部分領域：隣接した単位格子の集合によって構成される矩形領域．

部分領域境界：部分領域の外周．

カットライン：一つの部分領域を二つの部分領域に分割する水平または垂直の線分．図1の格子に沿って引かれる．

仮想ブロック：カットラインによって分割された二つの部分領域にまたがるネットの接続を保つために，カットライン上に置かれる仮想的なブロック．仮想ブロックは最終的にチャネル上に配置される．

部分領域境界のブロック集合：部分領域境界上のスロットまたはチャネルのいずれかに配置されることが決まった論理ブロックおよび仮想ブロックの集合．

部分領域内部のブロック集合：部分領域の内部のスロットに配置されることが決まった論理ブロック集合．

容量：カットラインと垂直に交差するトラックの数．

3.2 基本アルゴリズム

提案する配置概略配線手法は，レイアウト領域および論理ブロック集合を再帰的に分割することによる階層処理にもとづく．ここでは，単純かつ高速な領域の2分割処理を考える．

レイアウト領域をカットラインに沿って，二つの部分領域に分割することを考える．カットラインは図1の格子に沿って引くことにする．カットライン位置をこのようにとるのは，次のような理由による．第一に，カットラインとトラック（チャネル）が垂直に交差するため，カットラインを横切ることが可能なネット数を正確に表すことができる．FPGA のようにトラック数が前もって決まっている場合のレイアウト設計では，このような正確な見積もりが不可欠である．

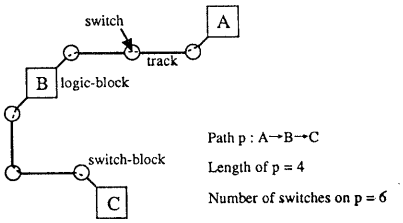


図 3 パス長
Fig. 3 Path length.

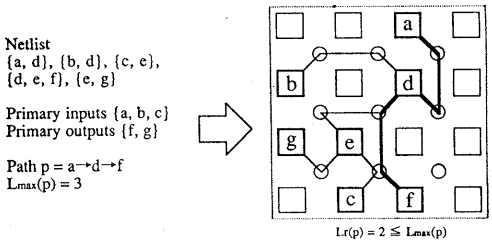


図 4 ネットリスト (左) およびレイアウト結果 (右)
Fig. 4 Netlist and its layout.

第二に，ネットがカットラインを横切る場合，その通過位置が，カットラインと垂直に交差するチャネル位置によって特定できる．したがって，領域の分割が進み，カットラインと交差するチャネルが唯一になれば，必然的にネットのチャネル位置が決定する．このとき，領域の分割により領域内のスロットも限定されるため，同時に，論理ブロックの配置位置も決まる．これは，このような再帰処理によって，論理ブロックの配置と概略配線（ネットのチャネルへの割当て）の同時処理の可能性を示唆しているといえる．

以上のような考えを基本として，これを実現する処理を提案する．

再帰処理を行うにあたり，まずI/Oブロックを**図5**左図に示すような領域の境界をなす4辺および4頂点に配置する．領域をカットラインに沿って二つの部分領域に分割すると，これに伴い対応するブロック集合 A, B, C は，分割された二つの部分領域のスロットおよびカットライン上のスロットに対応した三つのブロック集合にそれぞれ分割されることになる（図5右図）．このときカットラインによって分断された論理ブロック間に結線要求がある場合，ネットの接続を保つため，カットライン上に仮想ブロックを導入する．仮想ブロックはトラックに対応しており，その並びにより概略径路を特定できる．また，仮想ブロックは入出力端子を一つずつ持つ論理ブロックと考えられ，カットライン上の論理ブロックと仮想ブロックは等価なものとして以後の処理を行うことができる．つまり，配置処理と概略配線処理の同時処理が可能となる．

以下にアルゴリズムの概要を示す．

FPGA 配置概略配線アルゴリズム

Step 1. I/Oブロックの配置を行う．レイアウト領域全体を部分領域とみなし，待ち行列 Q に挿入．

Step 2. Q から部分領域 R を一つ取り出す．Q が空ならば終了．

Step 3. R をカットラインによって二つの部分領域 R_1 と R_2 に2分割する．

Step 4. R の分割に対応して，論理ブロック集合を図5に示すように分割する．

Step 5. カットライン上に割り当てられた論理ブロックに対して，それに接続するネットをカットラインに対しどちら側の端子に割り当てるかを決定する．

Step 6. カットラインと交差するネットは，カットライン上に仮想ブロックを生成する．

Step 7. R_1 および R_2 がさらに分割可能であれば，Q に挿入．

Step 8. Step 2 へ．

ここで Step 1 の I/O ブロックの配置は単純に，隣接する2辺に主入力を与える I/O ブロックを，その他の2辺に主出力を与える I/O ブロックを配置するものとする．また，カットラインは，できるだけ部分領域を正方形に近づけるため，部分領域の境界をなす辺のうち長い方の辺を分割するように定めるものとする．

以下，パス長制約を考慮したブロック集合の分割方法 (Step 4) および端子位置の割当て方法 (Step 5) について，水平方向のカットラインに関して述べる．

3.3 パス長制約にもとづいたブロック集合の分割

図5に示す論理ブロックおよび仮想ブロックの集合 A（左辺），B（内部），C（右辺）を分割する．分割は，まず部分領域境界のブロック集合 A, C から行い，続いて部分領域内部のブロック集合 B を行う．以下 A, C の分割が終了したと仮定し B の分割について述べる（部分領域境界のブロック集合の分割は後述する）．

3.3.1 部分領域内部のブロック集合の分割

図5の論理ブロック集合 B の各要素を，カットラインに対して上側，下側およびカットライン上に，スロット数を考慮して割り当てる．

ここでは，部分領域内部のブロック集合の要素 v にラベル $l(v)$ $(0 \leqq l(v) \leqq 1)$ を設定することを考える．ラベル $l(v)$ は，論理ブロック v がカットラインの上側，下側のうち，どちらに割当てが望まれるかを表す指標であり，1に近いほど上側，0に近いほど下側への割当てが期待されるように設定する．その値はブロック集合全体の接続を考慮したラベル $lc(v)$ と，パス長制約を考慮したラベル $lp(v)$ とを加え合わせることで算出する．このような設定を行うことで，ネットによって接続されている論理ブロック同士，およびパス長制約の与えられたパスに属す論理ブロック同士が，でき

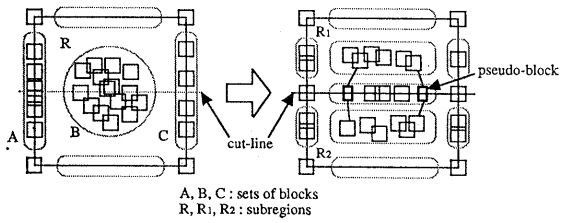


図 5 部分領域の分割による再帰処理
Fig. 5 Recursive process of partitioning a subregion.

るだけ同じ領域に割り当てられることを目指す．

ラベル $lc(v)$ は，次のように算出される．まず，部分領域Rの内部および境界上の論理ブロック，仮想ブロックを節点とする有向グラフ G_R を考える（以下論理ブロック（仮想ブロック）と節点は同一とみなす）．G_R の各枝は，最初に行われる再帰処理で，入力となるネットリストを出入力から主出力に向かって幅優先探索し，各ネットによって接続すべきブロックの集合において最も先に探索されたブロックから残りのブロックに対し有向枝を持つよう設定する．ここで設定される有向枝の向きは探索の順序を示しており，さらに幅優先探索によって，その順序は固定されている．つまり，G_R はサイクルのない有向グラフとなり，その枝はブロック間の結線要求を表すことになる．

R の部分領域境界のブロック集合（論理ブロックおよび仮想ブロック）の各要素に相当する節点vに対し，ブロックがすでに割り当てられた位置により，

$$lc(v)=\begin{cases}1 & (\text{カットラインの上側の節点})\\ 0 & (\text{カットラインの下側の節点})\\ 0.5 & (\text{カットライン上の節点})\end{cases}$$

というラベルを初期値として与える．このラベル値を有向グラフ G_R の枝の向きに沿って以下のように伝搬し，このときの各節点vのラベルを $lcf(v)$ とする．

ある節点vのラベル $lcf(v)$ は，vへの入枝の端点となっている節点の集合 $I(v)$ のラベルを用いて，

$$lcf(v)=\sum_{u\in I(v)} lcf(u)/|I(v)|$$

と設定する．つまり，$lcf(v)$ は $I(v)$ のラベルの平均値となる．こうすることで，各節点は隣接した節点のラベルに基づいたラベルをとることになり，ネットによる接続を考慮したラベル設定が実現できると考える．以下に lcf の算出アルゴリズムの概要を示す（**図6** 参照）．

ラベル lcf の算出アルゴリズム

Step 1. 部分領域境界のブロック集合に相当する節点に対し上記の初期値をラベルとして与え，これらの節点を待ち行列qに挿入．論理ブロック集合Bの要素である節点のラベルを0に初期化する．

Step 2. q から節点vを取り出す．q が空ならば終了．

Step 3. 節点 v の出枝の端点となっている節点の集合 $O(v)$ の要素であり，かつ論理ブロック集合Bの要素である節点uに対し，Step 4, Step 5 の処理を行う．

Step 4. ラベルの更新を行う．すなわち，

$$lcf(u)\leftarrow lcf(u)+[lcf(v)/|I(u)|]$$

Step 5. 節点 u に対し，$I(u)$ のすべての要素について Step 4 の処理（ラベルの更新）が終われば，u を待ち行列qに挿入．

Step 6. Step 2 へ．

有向グラフ G_R はサイクルを持たず，それゆえ，上記のアルゴリズムは，G_R のすべての枝を1度だけ探索して終了する．

同様に枝の向きと反対方向にラベルを伝搬する．このときの節点vのラベルを $lcb(v)$ とする．

各節点vのラベル $lc(v)$ は $lcf(v)$ と $lcb(v)$ とを平均することで算出する．

続いて，ラベル $lp(v)$ の算出を考える．パスがカットラインを通過する場合，そのたびごとに仮想ブロックが挿入され，パス長が1だけ増加する．したがって，パス長制約が課されたパスを構成する論理ブロックおよび仮想ブロックは，パスがカットラインと交差する必要がなければカットラインに対してできるだけ同じ側に割り当てることが望まれる．

考えている部分領域 R におけるパス p の形状は，**図7**（a）～（d）に示す4通りがある．このうち，図7（c）の場合はカットラインとの交差が避けられず，部分領域 R で必ずパス長の増加を起こす．また，図7（d）の場合はパスpに含まれる論理ブロックをカットラインのどちら側に割り当てるべきか，この段階で判断するのは適当でない．そこで，ここでは図7（a），（b）のような場合のみを考えることにする．

仮想ブロックはトラックに対応することから，これまでの再帰処理でパス p に挿入された仮想ブロックの数 $L(p)$ はこの段階でのパス長を示しているといえる．つまり，

$$s(p)=L_{\max}(p)-L(p)$$

とすれば，$s(p)$ は現段階でのパス長のスラック（余裕度）を表す．この値が小さいほど，パス p は厳しいパ

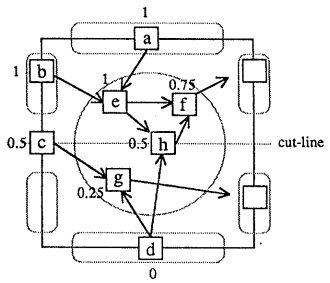


図6 ラベル lc の算出
Fig. 6 Calculation of label lc.

ス長制約が課されているといえる．そこでパス p を構成するブロックのうち部分領域 R の内部の論理ブロック v に対して，

$$c(p)=\pm 1/(s(p)+1)$$

を考える．符号は，カットラインの上側に v を割り当てたいとき正（+），下側のとき負（−）をとる．$c(p)$ は，$s(p)$ に反比例した値をとり，パス長制約が厳しいパスほど，すなわち $s(p)$ が小さいほど，その値は大きくなるよう定義されている．そこで，各論理ブロック v について，$c(p)$ の値を $lp(v)$ とし前述の $lc(v)$ と加え合わせれば，スラックの大きさにより $lp(v)$ もしくは $lc(v)$ が支配的となるラベル $l(v)$ を得られる．特に $s(p)=0$ ならば $lp(v)=\pm 1$ となり，これを $lc(v)$ に加えることで $lc(v)$ に関係なく，最終的なラベル $l(v)$ の値を 1（または 1 以上）もしくは 0（または 0 以下）に設定することができる．これは最優先でその論理ブロック v がカットラインの上側あるいは下側に割り当てられることを意味し，パス長制約を満足する割当てが期待できる．

実際には，論理ブロック v はパス p 以外でパス長制約の課されたパスに属すことがあるため，カットラインの上側に対する割当ての要求 $c(p)$ の最大値を $lp_1(v)$ $(0\leqq lp_1(v)\leqq +1)$，下側に対する割当ての要求 $c(p)$ の最小値を $lp_2(v)$ $(-1\leqq lp_2(v)\leqq 0)$ として，これらを加え合わせて $lp(v)$ とする．以下に lp の算出アルゴリズムの概要を示す．

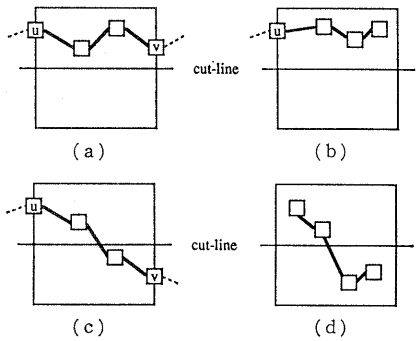


図 7 部分領域の内部でのパスの形状
（a）始点，終点がカットラインに対し同じ側の境界上，（b）始点または終点が境界上，（c）始点，終点がカットラインに対し異なる側の境界上，（d）始点，終点が部分領域の内部
Fig. 7 Path classification.
(a) Both start and end blocks of a path are on the boundary in the same side of the cut-line. (b) One of the start and end blocks of a path is on the boundary. (c) Start and end blocks of a path are on the boundary in the opposite sides of the cut-line. (d) Both start and end blocks of a path are inside the subregion.

ラベル lp の算出アルゴリズム

Step 1. 部分領域 R の内部の論理ブロック v について，$lp(v)=lp_1(v)=lp_2(v)=0$ とする．

Step 2. 部分領域 R の境界および内部のブロックで構成されるパスで，パス長制約の課されているものの集合を P とする．P に含まれる各パス p について Step 3〜Step 5 の処理を行う．P に含まれるすべてのパスについて処理が終われば，Step 6 へ．

Step 3. p が部分領域境界のブロック集合の要素を含まない場合（図 7（d）のような場合），Step 2 へ．

Step 4. p を構成するすべてのブロックをカットラインの上側に割り当てることが可能な場合（図 7（a），（b）のような場合），$c(p)=[+1/(s(p)+1)]$ とし，p を構成する各論理ブロック v について，

$$lp_1(v)\leftarrow \max\{lp_1(v), c(p)\}$$

またカットラインの下側に割り当てることが可能な場合，$c(p)=[-1/(s(p)+1)]$ とし，同様に，

$$lp_2(v)\leftarrow \min\{lp_2(v), c(p)\}$$

Step 5. Step 2へ．

Step 6. R の内部の論理ブロック v について $lp_1(v)=1$（または $lp_2(v)=-1$）のとき，

$$lp(v)\leftarrow lp_1(v)\ (\text{または } lp(v)\leftarrow lp_2(v))$$

そうでないとき，

$$lp(v)\leftarrow lp_1(v)+lp_2(v)$$

最終的なラベル $l(v)$ は，ラベル $lp(v)$ とラベル $lc(v)$ とを加え合わせて，

$$l(v)\leftarrow lc(v)+lp(v)$$

とする．ただし，$l(v)>1$ および $l(v)<0$ となった場合はそれぞれ $l(v)=1$，$l(v)=0$ とする．

以上の処理により得られたラベル値にもとづき，論理ブロック集合 B の各要素をカットラインの上側，下側，およびカットライン上に割り当てる．以下にラベル値にもとづく割当てアルゴリズムを示す．

割当てアルゴリズム

Step 1. 論理ブロック集合 B の各要素をラベル値をキーとして整列する．

Step 2. ラベル値が 0.5 を越える論理ブロックを，ラベル値の大きい順に，スロット数分だけカットラインの上側に割り当てる．

Step 3. ラベル値が 0.5 未満の論理ブロックを，ラベル値の小さい順に，スロット数分だけカットラインの下側に割り当てる．

Step 4. 以下の Case 1～Case 4 のいずれかを実行する.

Case 1. Step 2, Step 3 ですべての割当てが済んだとき，残りの論理ブロックをスロット数分だけカットライン上に割り当てる．さらに残った論理ブロックは，カットラインの上側，下側において，スロット使用率（＝割り当てられた論理ブロック数/スロット数）がなるべく等しくなるように割り当てる.

Case 2. Step 2, Step 3 の両方で割当ての済んでいない論理ブロックが残ったとき，残りの論理ブロックすべてをカットライン上に割り当てる.

Case 3. Step 2 で割当ての済んでいない論理ブロックが残り，かつ Step 3 の割当てはすべて済んだとき，残りの論理ブロックをラベル値の大きい順に，スロット数分だけカットライン上に割り当てる．さらに残った論理ブロックはカットラインの下側に割り当てる.

Case 4. Step 3 で割当ての済んでいない論理ブロックが残り，かつ Step 2 の割当てはすべて済んだとき，残りの論理ブロックをラベル値の小さい順に，スロット数分だけカットライン上に割り当てる．さらに残った論理ブロックはカットラインの上側に割り当てる.

このような処理により，ラベル値が1に近い論理ブロックほどカットラインの上側に，ラベル値が0に近い論理ブロックほどカットラインの下側に割り当てられ，部分領域内部のブロック集合の分割が終了する.

3.3.2 部分領域境界のブロック集合の分割

図5において部分領域境界のブロック集合 A, C の分割を考える（A, C は仮想ブロックをも含む）．分割は A, C の順に逐次的に行うものとする．ここでは A の分割に注目して述べる.

ブロック集合 A に関しても，3.3.1 項と同様なラベル付けおよび分割を考えることができる．しかし，部分領域境界上の理論ブロック（仮想ブロック）は，通常，隣接する二つの部分領域のインタフェース部分となるため，部分領域 R に対して左に隣接する部分領域 R_L を考え，R と R_L を結合した領域（結合領域と呼ぶ（**図8**））を処理の対象とする（ただし，隣接する部分領域が存在しない場合，R のみを処理の対象とする）．ラベル付けは，結合領域の内部の論理ブロック（仮想ブロック）に対して前項同様に行う．ラベル lc の算出の際，結合領域の右辺および左辺のブロック集合の分割が終了していない場合は，これらを考慮しないことにする．すなわち結合領域の上辺および下辺（および分割の終了している右辺，左辺）がラベルの伝搬の対象となる．同様にラベル lp も算出することができる.

そして，ブロック集合 A を構成する各要素を，ラベル値にもとづき，カットラインの上側，下側，およびカットライン上に割り当てる．仮想ブロックはチャネルに割り当てる必要があり，そのため，論理ブロックとは別個にチャネルを構成するトラック数分だけ，チャネルに割当て処理を行うものとする.

以上のような方法により部分領域境界のブロック集合の分割が終了する．なお，ブロック集合 A は部分領域 R の左辺であるのと同時に，部分領域 R_L の右辺でもあるため A の分割により R_L の右辺の分割も終了したことになる.

3.4 パス長制約にもとづいた端子割当て

対象とする論理ブロック（図2）は，入力端子 A～D が機能的に等価な端子であり，どの端子にどのネットを接続するかを決定する必要がある．そこで，カットライン上に割り当てられた論理ブロック B に接続するネットを，カットラインに対して B のどちら側の端子に割り当てるかを決定する.

まず，端子の割当て対象の論理ブロックを含む．パス長制約の課されたパスを考える．そして，このようなパスの一部を構成するネットを，パスのスラックの昇順（ネットが複数のパスの一部となっている場合，最も制約の厳しいパスを採用する）に，割当て可能な端子数分だけ，そのパスとカットラインとが交差しないような端子に割り当てることにする（**図9**（a）のネット a, b）．ここで，割当て可能な端子とは，ネットが論理ブロックの入力を与える場合は入力端子，出力となる場合は出力端子を意味する（複数の端子が割当ての候補になる場合，その中のどの端子に割り当てるかは後に続く再帰処理によって決定する）．また，このとき割り当てられなかったネットは，図9（a）に示す

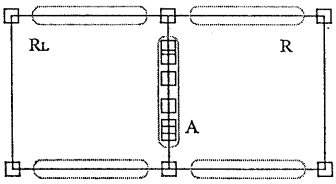


図8 結合領域
Fig. 8 Augmented region.
A: a set of blocks to be partitioned, R: subregion, R_L: subregion adjacent to R.

ネット c のように，反対側の端子に割り当てる．

一方，パス長制約の課されたパスを含まない通常のネットは次のような端子の割当てを行う．まず，図9(b)に示すネット d, e のように，B を除いて，接続すべきすべての論理ブロックがカットラインの片側にあるネットを割当て可能な端子数分だけ，ネットが接続すべき論理ブロックと同じ側にある端子に割り当てることにする．このとき割り当てられなかったネットは，ネット f のように，反対側の端子に割り当てる．

さらに，図9(c)に示すネット g のように，カットラインの両側に接続すべき論理ブロックがあるネットは，どのような端子の割当てを行ってもカットラインとの交差が生じる．そこで，図9(b)の場合の処理に続き，カットラインに対して割当て可能な端子が存在する側の端子にネットを割り当てる処理を行う．

このような割当て処理によって，パス長制約の課されたパスは，優先的にカットラインと交差しないような端子の割当てを達成できる．通常のネットは不必要なカットラインとの交差を回避できる．論理ブロックの分割および端子割当ての結果，カットラインと交差する各ネットに対しては，一つの仮想ブロックをカットライン上に設け，ネットの接続を保持する．なお，生成した仮想ブロックの数がカットラインの容量を越えた場合，オーバフローとなり，未結線が生じることになる．

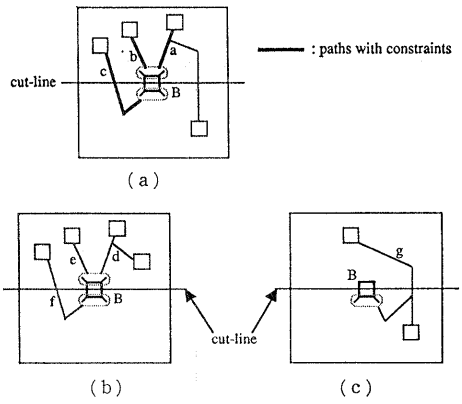


図 9 端子の割当て
(a)パス長制約を持つネット，(b)カットラインの片側に接続すべきブロックがあるネット，(c)カットラインの両側に接続すべきブロックがあるネット
Fig. 9 Assignment of nets to terminals.
(a) Nets with path length constraints.
(b) Nets whose blocks are on one side of the cut-line. (c) Nets whose blocks are on both sides of the cut-line.

4. 計算機実験による手法の評価

提案手法を SUN SPARC station 2 (28.5 MIPS) 上にC言語で実装し，計算機実験によって評価した．データとして，MCNC ベンチマーク回路を mis-pga[12] と呼ばれるテクノロジーマッパーによって，FPGA の論理ブロックにマッピングすることで得た回路を用いている．

FPGA の配置配線設計において，タイミングやパス長を制御する手法はこれまで文献による発表は行われておらず，ここでは，提案手法においてパス長制約を考慮しない場合，およびパス長制約を考慮した場合について，パス長の比較実験を行った．パス長制約は，入力回路において論理ブロックの段数が最大となるすべてのパスに対し，文献 1) にならい，入力回路をパス長制約なしで配置配線した場合の最大のパス長の 70% を L_{max} として与えた．

さらに，提案手法の配置概略配線における基本的な性能（すべての配線を行うのに要する総配線長およびトラック数）を客観的に評価するため，従来通りの段階的処理による配置・概略配線手法による結果との比較を行った．配置処理にはシミュレーティッドアニーリング法 (SA 法)[10] によるペア交換配置，および min-cut 配置手法[2] を用い，概略配線には文献 11) の階層的概略配線手法を用いた．目的関数はいずれも総配線長の最小化である．

表 1 に回路の諸元，**表 2～表 4** に実験結果を示す．表1において制約パス数とは制約を課したパスの数を表す．表2，表3において最大パス長とは，制約を課したすべてのパスに対して，そのパス長の最大値を表す．表4において従来手法の実行時間は配置に要する時間と配線に要する時間の和の形で示す．表2～表4

表 1 ベンチマーク回路
Table 1 Benchmark circuits.

回路名	論理ブロック数	ネット数	端子数	制約パス数	L_{max}
con1	15	13	35	14	6
rd53-hdl	24	21	55	4	14
rd53	26	23	87	3	10
misex1	42	35	133	29	14
z4ml	53	49	197	7	20
f51m	87	79	319	12	25
misex2	126	108	361	6	10
bw	135	107	475	14	19
vg2	258	250	979	105	40
duke2	413	384	1559	14	50

においてトラック数とは，すべてのネットを概略配線するのに必要な1チャネル当たりのトラック数，総配線長とは，そのとき概略配線に要した総トラック数を表す．概略配線は，ネットに対し径路のガイドラインを与えているといえる．そのため，詳細配線後の実配線長として，本実験で用いた総配線長とほぼ同等の結果を得ることが期待できる．

図 10 に提案手法による回路 vg 2 の出力結果を示す．図 10 において実線の矩形が論理ブロック，点線の矩形がスイッチブロックを表す．太実線の矩形は論理ブロックが実際に配置されたスロットを表している．同時にいくつかのネットの概略径路を表示した．

表2，表3を比較すると，パス長制約を考慮することで，ほぼすべての場合について，制約違反パスを0にすることができた．この場合に総配線長およびトラ

表 2 パス長制約を考慮しない場合の実験結果
Table 2 Experimental results without path length constraints.

回路名	違反パス数	最大パス長	トラック数	総配線長	処理時間［s］
con1	5	8	2	18	0.05
rd53-hdl	3	20	2	42	0.09
rd53	3	15	3	81	0.15
misex1	8	21	4	117	0.28
z4ml	7	29	5	235	0.45
f51m	4	36	6	383	0.56
misex2	2	14	5	490	0.68
bw	5	28	6	528	0.67
vg2	12	57	10	1941	2.51
duke2	14	75	12	3187	4.61
計	63	303	55	7022	10.05

表 3 パス長制約を考慮した場合の実験結果
Table 3 Experimental results with path length constraints.

回路名	違反パス数	最大パス長	トラック数	総配線長	処理時間［s］
con1	0	6	2	20	0.07
rd53-hdl	0	13	2	49	0.10
rd53	0	10	3	77	0.15
misex1	0	14	4	127	0.38
z4ml	0	20	5	250	0.48
f51m	0	25	6	398	0.87
misex2	1	11	5	528	0.88
bw	0	19	6	556	1.29
vg2	1	42	11	2122	6.15
dube2	0	50	12	3128	6.74
計	2	210	56	7255	17.11

表 4(a) 従来手法による実験結果（SA 法によるペア交換配置＋階層的概略配線手法）
Table 4(a) Experimental results (SA placement +hierarchical global routing).

回路名	トラック数	総配線長	処理時間［s］
con1	2	28	28+ 0.09
rd53-hdl	2	42	39+ 0.12
rd53	3	85	156+ 0.21
misex1	4	148	248+ 0.29
z4ml	5	248	699+ 0.77
f51m	6	408	998+ 1.05
misex2	5	490	4372+ 1.20
bw	6	699	10742+ 1.57
vg2	11	2274	34369+ 6.44
duke2	14	3604	161446+13.83

表 4(b) 従来手法による実験結果（min-cut 配置＋階層的概略配線手法）
Table 4(b) Experimental results (min-cut placement+hierarchical global routing).

回路名	トラック数	総配線長	処理時間［s］
con1	4	37	0.05+ 0.05
rd53-hdl	3	48	0.09+ 0.12
rd53	5	94	0.22+ 0.24
misex1	6	145	0.44+ 0.39
z4ml	6	279	0.90+ 0.55
f51m	8	465	3.11+ 0.77
misex2	9	715	6.88+ 1.38
bw	8	687	8.81+ 1.07
vg2	15	2594	44.41+ 4.22
duke2	16	3981	169.58+15.62

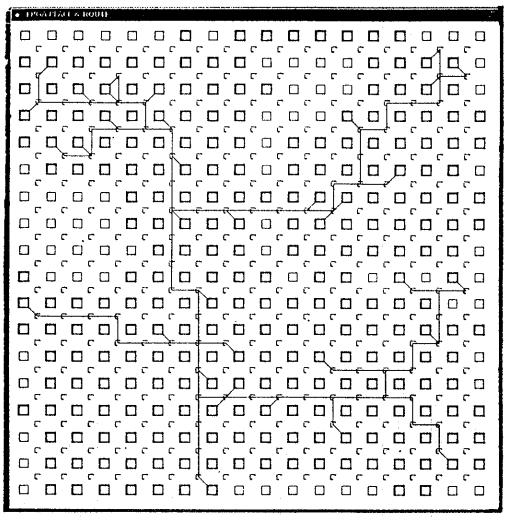


図 10 提案手法による配置概略配線結果（回路 vg2）
Fig. 10 Layout result for vg2.

ック数の増加は平均2〜3%程度である．また，処理時間はパス長制約を考慮しないときに対して平均1.7倍程度かかるが最大でも数秒に抑えられている．

さらに，表4との比較を行うと，提案手法は，単独の配置問題，配線問題に対して，良質の解を出すことで知られているSA法によるペア交換配置および階層的概略配線と，基本的な性能に関して同等以上の結果を示している．また，高速な処理を実現するmin-cut配置手法を用いた場合よりもさらに高速な処理を行っていることがわかる．

従来手法に比較して，良好な結果を得た理由は，次のように考えることができる．従来の配置処理では，単に論理ブロック間の抽象的な結線要求にのみ影響を受けて，これらの論理ブロックをできるだけ近づけて配置するため，実際の概略配線を考慮せず，結果として概略径路の迂回やトラック数の増加を起こす可能性が高い．一方，提案手法では，処理の進行に伴って，論理ブロック間の結線要求は具体的な概略径路（仮想ブロックによって表されている）に変換される．以後，仮想ブロックおよび論理ブロックに対する結線要求を同等に扱うことで，概略径路を意識した処理を行うことが可能となる．

例えば，**図11** 左のような結線要求のある論理ブロックを考える．従来手法では論理ブロックを結線要求に基づき近づけて配置するが，必ずしも仮想配線どおりに概略配線されるとは限らず，トラック数の影響で概略径路の大きな迂回を招くことがあり得る（図11（a））．一方，提案手法では仮想ブロックが導入され，従来手法と反対側の結線要求が生じ，より短い概略径路を生成することが可能となる（図11（b））．

以上より，提案手法の有効性を実験的に確認することができた．

5. むすび

本論文では，パス長制約を考慮したFPGA配置概略配線同時処理手法を提案した．計算機実験の結果，パス長の制御に関する有効性および配置配線処理における基本的な性能に関する有効性を確認した．

今後の課題として，パス長の制御の際に，文献9）にあるようなdirect interconnectionおよびスイッチブロックを経由せずに論理ブロック内を通過するパスを有効に利用すること，さらにテクノロジーマッピング等の上位設計工程の処理を本手法に統合していくことがあげられる．

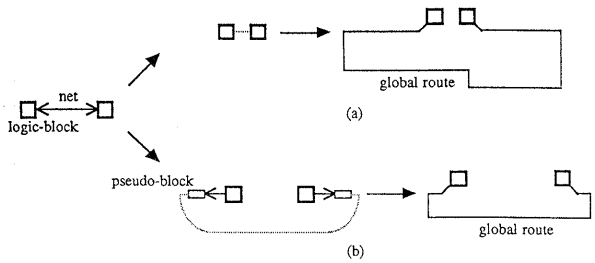


図11 従来手法と提案手法の比較
（a）従来手法，（b）提案手法

Fig. 11 Comparison of conventional methods with the proposed algorithm.
(a) conventional methods, (b) proposed algorithm.

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いただきました，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．

参考文献

1) 安藤 宏：性能指向レイアウトCADと適用事例，第8回ADEE・ジャパン'93セミナーA10資料 pp. 21-30 (1993).
2) Breuer, M. A.: Min-Cut Placement, *J. Design Autmation and Fault Tolerant Computing*, Vol. 1, No. 4, pp. 343-362 (1977).
3) Brown, S. D., Francis, R. J., Rose, J. and Vranesic, Z. G.: *Field-Programmable Gate Arrays*, Kluwer Academic Publishers, Boston (1992).
4) Burstein, M., Hong, S. J. and Pelavin, R.: Hierarchical VLSI Layout: Simultaneous Placement and Wiring of Gate Arrays, *Proc. VLSI '83*, pp. 45-60 (1983).
5) Dai, W.-M. and Kuh, E. S.: Simultaneous Floor Planning and Global Routing for Hierarchical Building-Block Layout, *IEEE Trans. Comput.-Aided Des. Integrated Circuits & Syst.*, Vol. CAD-6, No. 5, pp. 828-837 (1987).
6) Dunlop, A. E., Agrawal, V. D., Deutsch, D. N., Jukl, M. F., Kozak, P. and Wiesel, M.: Chip Layout Optimization Using Critical Path Weighting, *Proc. 21st DA Conf.*, pp. 133-136 (1984).
7) El Gamal, A., 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*, Vol. 24, No. 2, pp. 394-398 (1989).
8) Hsieh, H.-C., Carter, W. S., Ja, J., Cheung, E.,

Schreifels, S., Erickson, C., Freidin, P., Tinkey, L. and Kanazawa, R.: Third-Generation Architecture Boosts Speed and Density of Field-Programmable Gate Arrays, *Proc. IEEE 1990 Custom Integrated Circuits Conf.*, pp. 31.2.1-31.2.7 (1990).

9) Kawana, K., Keida, H., Sakamoto, M., Shibata, K. and Moriyama, I.: An Efficient Logic Block Interconnect Architecture for User-Reprogrammable Gate Array, *Proc. IEEE 1990 Custom Integrated Circuits Conf.*, 31.3.1-31.3.4 (1990).

10) Kirkpatric, S., Gelatt, C. D., Jr. and Vecchi, M. P.: Optiimzation by Simulated Annealing, *Science*, Vol. 220, No. 4598, pp. 671-680 (1983).

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

12) 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).

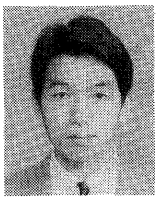
13) Muroga, H., Murata, H., Saeki, Y., Hibi, T., Ohashi. Y., Noguchi, T. and Nishimura, T.: A Large Scale FPGA with 10K Core Cells with CMOS 0.8 μm 3-Layered Metal Process, *Proc. IEEE 1991 Custom Integrated Circuits Conf.*, pp. 6.4,1-6.4.4 (1991).

14) Ohmura, M,. Wakabayashi, S., Toyohara, Y., Miyao, J. and Yoshida, N.: Hierarchical Floorplanning and Detailed Global Routing with Routing-Based Partitioning, *Proc. 1990 IEEE Int. Symposium on Circuits and Syst.*, pp. 1640-1643 (1990).

15) Suaris, P. R. and Kedem, G.: A Quadrisection-Based Combined Place and Route Scheme for Standard Cells, *IEEE Trans. Comput.-Aided Des. Integrated Circuits & Syst.*, Vol. 8, No. 3, pp. 234-244 (1989).

16) 寺井正幸，八原俊彦：タイミング駆動型ミニカット配置アルゴリズム，信学論(A)，Vol. J75-A, No. 6, pp. 1054-1063 (1992).

(平成5年7月29日受付)
(平成6年1月13日採録)



戸川　望（学生会員）

昭和45年生．平成4年早稲田大学理工学部電子通信学科卒業．現在，同大大学院修士課程在学中．電子回路の設計自動化，計算幾何学，グラフ理論等の研究に従事．電子情報通信学会会員．

佐藤　政生（正会員）

昭和34年生．昭和58年早稲田大学理工学部電子通信学科卒業．昭和58年同大大学院博士前期課程修了．昭和61年同後期課程修了．工学博士．昭和59年早稲田大学情報科学研究教育センター助手．昭和61年カリフォルニア大学バークレー校研究員．昭和62年拓殖大学工学部情報工学科助教授を経て，現在早稲田大学理工学部電子通信学科助教授．電子回路の設計自動化，ノイズ解析，計算幾何学，グラフ理論等の研究に従事．昭和62年度丹波記念賞受賞．平成2年安藤博学術奨励賞受賞．IEEE, ACM, 電子情報通信学会，プリント回路学会，日本 OR 学会各会員．

大附　辰夫（正会員）

昭和15年生．昭和38年早稲田大学理工学部電気通信学科卒業．昭和40年同大大学院修士課程修了．同年日本電気(株)入社．昭和55年同退社．現在，早稲田大学理工学部電子通信学科教授．工学博士．電子回路の CAD およびこれに関連した基礎研究に従事．昭和44年度電子情報通信学会論文賞受賞．1974年 IEEE CAS Society より Cuillmin-Cauer 賞受賞．共著「VLSI の設計 I」(岩波書店)，編共著「Layout Design and Verification」(North-Holland)．電子情報通信学会，電気学会，プリント回路学会各会員．IEEE Fellow.