低消費電力を指向した パフォーマンスドリブン配置手法とその評価

正 員 吉川 雅弥* 正 員 寺井 秀一*

Performance Driven Placement Procedure for Low Power

Masaya Yoshikawa*, Member, Hidekazu Terai*, Member

Deep-Sub-Micron (DSM) technologies of $0.18\mu\text{m}$ and below enable the integration of logical circuits having more than 10 million gates. In such a DSM technology, it's important to consider reducing power consumption, improving interconnection delay, and dispersing wire congestion at initial phase of layout design. In this paper, we proposed a novel performance-driven placement algorithm. The proposed algorithm based on Genetic Algorithm(GA) has a two-level hierarchical structure. For selection control, new objective functions are introduced for reducing power consumption, improving interconnection delay and dispersing wire congestion. Studies on floor planning and cell placement have been reported as being applications of GA to the LSI layout problem. However, no studies have ever seen the effect of applying GA in consideration of power, delay and congestion. Results show improvement of 11.7% for the total wire length of the nets with high SW rate, 22.5% for the worst path delay and 15.9% for wire congestion on average.

キーワード：低消費電力，配線遅延，配線混雑，進化戦略，LSI レイアウト設計

Keywords: Low power, interconnection delay, Wire congestion, Evolutionary method, LSI layout design

1. はじめに

デシミクロンオーダーの VLSI 設計においては，タイミング，消費電力，配線混雑などをトータルで最適化するフィジカルデザイン手法が求められる。ここで，タイミングや配線混雑を最適化するためにレイアウト結果を修正する従来の対応方法では，対策箇所が数十～数百件/Chip までが限界である。一方， $0.1\mu\text{m}$ 以下のデシミクロンプロセスでは，要対策箇所は数千～数万件/Chip となることが予測される⁽¹⁾。そのため，チップの最適なレイアウトを実現するには，従来，個別に評価していた制約モデルを統合し，互いに背反する要因間のトレードオフのもとで最適な解を導く新しいレイアウトアルゴリズムが必要になってくる。

また，SIA（米国半導体協会）のロードマップによれば，2014 年にはデザインルールが $0.035\mu\text{m}$ ，MPU におけるロジック部のトランジスタ数は 40 億超，チップ面積が $25\text{mm} \times 25\text{mm}$ になると予測している⁽¹⁾。これを仮に，現時点のプロセステクノロジーと設計技術を用いて実現するとす

れば，1 辺が $75 \sim 80\text{mm}$ の巨大チップを設計することに相当する。このことは，今後さらにチップ消費電力とタイミング制約がより深刻な設計課題になってくることを如実に物語っている。

そこで本研究では，消費電力やタイミングを総合的に考慮した配置手法を提案する。提案手法の基本アルゴリズムは，遺伝的アルゴリズム（Genetic Algorithms: GA）⁽²⁾⁻⁽¹²⁾ であり，筆者らは文献 (12) において，GA をセル配置問題に適用し，動作原理としての有効性を確認している。セルとは AND/OR などの基本ゲートやフリップフロップなど比較的小規模の論理エレメントであり，LSI レイアウト設計の単位要素である。また，GA は生物の進化の過程を工学的にモデル化したアルゴリズムであり，選択・交叉・突然変異という 3 つの遺伝オペレーションを適用することにより世代交代を繰り返し，集団を進化させ解を導く。GA を LSI レイアウト問題に適用した報告として，フロアプラン⁽⁵⁾⁽⁶⁾やチャネル配線⁽⁷⁾⁻⁽¹¹⁾に関する報告があるが，本論文で提案する低消費電力を指向したパフォーマンスドリブン配置手法は新しいアプローチである。

2. 基本アルゴリズム

本論文では CMOS スタンダードセルを対象に図 1 に示す配置モデルを用いた。列を $Width_{min}$ （全セルの中で $Width$

* 立命館大学 総合理工学研究機構 VLSI センター

〒 525-8577 滋賀県草津市野路東 1-1-1

VLSI center, Research Organization of Science and Engineering, Ritsumeikan University

1-1-1, Nojihigashi, Kusatsu, Shiga-ken, 525-8577

図 1 配置モデル

Fig. 1. Placement model.

図 2 階層化配置手法

Fig. 2. Hierarchical placement.

が最も小さいもの), 行を $Channel/2 + Height + Channel/2$ で区切った配置メッシュを定義し, 各配置メッシュにセルを配置する。また, 本研究で提案する配置手法は, 概略配置と詳細配置の 2 ステップで構成した。

まず概略配置では, 図 2(1) に示すように, 配置領域を 4 つに分割し, 各分割領域ごとにセルの割り振りを行い, これを繰り返すことによって配置を段階的に決定していく。すなわち, セルを 4 つのグループに分け, 各グループを LSI チップの 4 つの領域にそれぞれ対応させる。各部分領域とそのセルグループに対して, 上と同様の 4 分割処理を逐次繰り返し, 部分領域とセルグループを細かくしていく。後段の詳細配置に比べて解空間が狭くなるために探索に要する時間は短くなる。概略配置がチップ領域の多段分割を繰り返すことによって配置を求める手法であるのに対し, 詳細配置では, 図 2(2) に示すように, LSI 上に 2 次元グリッドを設定し, 概略配置の最終段階で得られている各部分領域内のセルの最適な配置位置を決定する。解空間が広い場合探索には時間がかかるが, 配置領域を直接扱うため探索が十分に行われた場合には, 最適な解を取得できる可能性が高い。

3. コーディング

最適化問題に GA を適用するには, 対象となる問題のモデル化が必要である。概略配置でのコーディングは, 図 3

図 3 概略配置のコーディング

Fig. 3. Coding for outline placement.

図 4 詳細配置のコーディング

Fig. 4. Coding for detail placement.

のように行った。図 3 は概略配置の途中のステップ, 8 個のセルを A から D の 4 つの領域に分割した例を示したものである。遺伝子座にセル ID を, 遺伝子に配置領域名を対応させる。すなわち, セル ID 順に分割領域名を保存する形で個体 (1 つの個体は, 1 つの配置パターン) を表現する。図 3 では, '2' と '8' のセルが Area-A へ, '5' と '6' が Area-B へ分割されている。一方, 詳細配置でのコーディングは, 図 4 のように行った。図 4 は, 5 個のセルの配置例を示したもので, 遺伝子座にセル ID を, 遺伝子に左端の配置グリッド名を対応させる。

4. 選択オペレーション

選択オペレーションは, 評価が高い個体 (適応度の高い個体) が次世代により多くの子孫を残すという自然淘汰の考えに基づいている。本研究では, 以下に示す 5 つの評価関数を導入した。

4.1 チップ面積による評価 面積の最小化は, LSI の性能やコスト, 製造歩留まり (良品率) をよくするため重要な指標になる。チップ面積は個々のセル面積の総和と配線に必要なスペースを足し合わせたものである。セルの面積は論理設計の結果によって使用するセルの種類が決められているため変更することができず, 配線スペースを小さくすることが配置設計での評価となる。

ここで, 配線スペースと配線の長さには相関があることが知られており⁽¹⁹⁾, 配線スペースの基準として, 一般に配線の長さの総和を用いる。しかし, 配置の段階ではネット (同じ信号で接続される端子の集合) の配線経路は決定していないため, 配線長を推定によって求めなければならない。従来から配線長の推定値として用いられているものに, 図 5(2) に示すようなネットの最小外接矩形の半周長がある。

図5 配線長の見積もり法

Fig. 5. Estimate method for virtual wire length.

しかし、この値は簡単に求められるが、実際の配線長の下限を与えているに過ぎず、特に多端子ネットの場合に実際の配線との誤差が大きくなる。また、別の推定方法として、図5(3)に示すようなスタイナー木⁽¹³⁾がある。

スタイナー木は、端子以外の場所で分岐を許した木であり、木の全長が最少のものが最小スタイナー木である。スタイナー木は実際の配線との誤差は小さいが、非常に多くの計算時間を要する。そこで、本研究では多端子の影響を取り入れ、かつ高速に計算できる方法として、最小外接矩形の半周長に補正を加える方法を用いる。今、 n 個の端子のネット i において、 i の最小外接矩形の半周長を $l_b(i)$ とし、 i を結ぶ最小スタイナー木の長さ $l_s(i)$ の上限値を $p_n l_b(i)$ とする。すなわち、

$$l_b(i) \leq l_s(i) \leq p_n l_b(i) \quad \dots\dots\dots (1)$$

である。 p_n は、 $n > 10$ の場合、 n での漸近形

$$p_n \simeq \frac{\sqrt{n} + 1}{2} \quad \dots\dots\dots (2)$$

が文献(14)で求められている。本研究ではネット i の仮想配線長 $length(i)$ を式(3)のように $l_b(i)$ と $p_n l_b(i)$ の平均値とし、チップ面積による評価をすべてのネットの仮想配線長の総和とした。

$$length(i) = \frac{l_b(i) + p_n l_b(i)}{2} = \frac{\sqrt{n} + 3}{4} l_b(i) \quad \dots (3)$$

4.2 配線混雑による評価 配線が混雑すると、未配線ネットが生じたり、配線スペースを増やすことによるチップサイズの増大を引き起こす。そこで、配線混雑を抑制するための評価として、分割を繰り返していく各過程での各領域(図6では、Area-A, B, C, D)をまたがるカットラインの中で最大のカットライン数から最小のカットライン数を引いた値(以後、カットライン偏差と呼ぶ)を用いる。図6は、ある分割におけるカットライン偏差を示したもので、(1)のカットライン偏差は3であり、(2)のカットライン偏差は0である。このカットライン偏差が小さいほど配線の集中を抑えることができ望ましい分割であると言える。また、図6における対角領域(Area-AとD, Area-BとC)にまたがるネットは、1つのネットにつき、カットライン偏差を1つ加算することとした。このように多段分割処理を繰り返すことによって各領域をまたがるカットライン数の差が平準化され、結果として配線混雑の分散化が図られることになる。

図6 配線混雑

Fig. 6. Wire congestion.

4.3 タイミング制約による評価 0.18 μ m以下のデシクロンプロセスでは、論理設計の段階でのタイミング制約をレイアウト設計で満たせないタイミング・クロージャの問題⁽¹⁷⁾が顕在化してきた。この問題を回避するため、以下に示す評価関数を新しく考案した。

4.3.1 配線遅延モデル 配線遅延の計算を評価演算の中で行うため、解の精度と処理時間の観点から、配線層は2層配線とし、第1層を垂直配線に、第2層を水平配線に使用するものとする。配線の等価回路をRC集中回路と仮定し、配線遅延モデルはElmore Delay⁽¹⁵⁾を用いる。多端子ネット n_i をスタイナー木で表した場合、Elmore Delayの上限値

$$d_{max}(i) = (R_0 + rL_k)(cW + \sum_j C_j) \quad \dots\dots\dots (4)$$

が文献(16)で求められている。

ここで、 R_0 はソースの出力等価抵抗、 L_k はソースからシンク k までの配線長、 r と c はそれぞれ単位長あたりの配線抵抗と配線容量である。また、 W はスタイナー木の配線長、 $\sum_j C_j$ はシンク容量の総和である。本論文では、仮想配線長を4.1節で述べたように見積もるため、ネット i のソースからシンク k までの配線遅延時間 $d_k(i)$ は、

$$d_k(i) = (R_0 + r_1 l_v + r_2 l_h) \times (c_1 L_v + c_2 L_h + \sum_j C_j) \quad \dots\dots\dots (5)$$

とする。ただし、 R_0 はネット i に入力するセルの出力等価抵抗、 $\sum_j C_j$ はネット i のシンク容量の総和、 $r_1 \cdot r_2$ は第1,2層の単位長あたりの配線抵抗、 $c_1 \cdot c_2$ は第1,2層の単位長あたりの配線容量を表す。また図7に示すように、 $l_v \cdot l_h$ はソースからシンク k のそれぞれ第1,2層の配線長、 $L_v \cdot L_h$ はネット i のそれぞれ第1,2層の配線長である。

4.3.2 タイミング制約 タイミング制約を満足するためには、全てのパスディレイが1クロックサイクル以内に収まる必要がある。ここで、パスディレイとは、図8に示すように、フリップフロップ(以下、FF)からFFまでのパスに沿った信号伝播時間のことである。図8の場合、パスは全部で2本あり、太線のパスディレイ T_d は、以下のように求めることができる。

$$T_d = g_1 + i_1 + g_2 + i_2 + g_3 \dots \dots \dots (6)$$

ただし, g_1, g_2, g_3 は,それぞれ FF1, Cell-1, FF2 のゲート遅延である。また, i_1, i_2 は,それぞれ FF1 から Cell-1 までと Cell-1 から FF2 までの配線遅延である。ここで, 全てのパスディレイを毎世代・全個体について計算すると, 計算量が増大する。そこで, 本研究では処理時間の短縮を目的に, 図 9 に示すようにある特定の世代の時のみ全てのパスディレイを計算・評価し, それ以外の世代では, 特定のパスについてのみパスディレイの計算・評価を行った。具体的な処理手順を以下に説明する。

- Step1: 0 世代目(初期配置)の各パスの遅延を計算する。
 Step2: Step1 で計算したパスで, 遅延が大きいものから Q 本のパス(Q はパラメータ)をパスリストとする。
 Step3: T 世代まで, パスリストのパスの遅延の合計値をタイミング制約による評価とする。
 Step4: $T+1$ 世代目まで全てのパスの遅延を計算し, Step2 と同様に新しいパスリストを作成する。

図 7 配線遅延のための仮想配線長

Fig. 7. Virtual wire length for interconnection delay.

図 8 パスディレイ

Fig. 8. Path delay.

図 9 パスディレイ計算のタイミング

Fig. 9. Timing of calculation for path delay.

Step5: 終了世代まで, Step3・4 を繰り返す。

このように提案手法では, “世代交代を繰り返して, 解を導く” という GA の特性を考慮して, クリティカルパス(回路の動作を決定するパス)を近似することで, 高い精度を保ちつつ処理時間を短縮することができる。

4.3.3 IR ドロップを考慮したゲート遅延 各セル列に電源を供給する電源リングから遠い位置に配置したセルは, IR ドロップ(電源配線上に生じる電流 I と配線抵抗 R の積の電圧降下)によって, ゲート遅延が増大する。そこで IR ドロップによるゲート遅延の増加を考慮するために, 図 10 に示すように配置領域をゾーンで区切り, 電源リングから遠いゾーン(図 10 の場合は, $zone2 \cdot 3$)に配置したセルのゲート遅延に重みを加えた。すなわち, セル i のゲート遅延 $g(i)$ は,

$$g(i) = \alpha \times g_{worst}(i) \dots \dots \dots (7)$$

とした。ただし, α は重みパラメータ, g_{worst} はセル i のワーストケースの値である。

4.4 低容量化による評価 スケーリング則⁽¹⁸⁾に代表される微細化技術により, 消費電力は 3 年で 4 倍に増加している。しかし, 発熱や実装, 携帯機器でのバッテリーの観点から低電力化が必須となってきた。そこで, 低消費電力を実現するために, 新しい評価関数を考案した。CMOS LSI の消費電力は次の 3 つの合計である。

- ① 信号線のスイッチングによる消費電力
- ② セル内の貫通電流による消費電力
- ③ リーク電流による消費電力

ここで, ①の消費電力 P_1 は消費電力全体の約 70% を占め, 式 (8) のように求めることができる。

$$P_1 = \beta \times C \times f \times V^2 \dots \dots \dots (8)$$

ただし, β は活性化率(信号の遷移確率), C は負荷容量(次段のゲート入力, このゲートの出力および配線の容量), f は周波数, V は電源電圧である。ここで消費電力の削減を考えた場合, ①の消費電力を削減することが有効である。そこで, β が大きいセル(活性化率が高いセル)の負荷容量 C を下げる(配線の負荷容量を下げる)ために, 活性化率が閾値以上のネットの仮想配線長を優先的に短くする評価関数 P_{length} を導入した。

図 10 IR ドロップを考慮したゲート遅延

Fig. 10. Gate delay for IR drop.

$$P_{length} = \sum_i \beta(i) \times length(i) \dots\dots\dots (9)$$

ただし、 i は活性化率が閾値以上のネット、 $\beta(i)$ はネット i の活性化率、 $length(i)$ は式 (3) での仮想配線長である。

4・5 消費電力分散による評価 消費電力の分散化に関して、消費電力の大きなセルを 1 箇所に配置すると、発熱などの弊害を引き起こす可能性がある。また、1 つのセル列 (同じ電源ラインで駆動するセル列) に同時にスイッチングする消費電力の大きなセルを配置するとピーク電流が増大する。そこで概略配置ステップでは、図 11 に示すようにチップ全体での消費電力の分散化をはかるために、各分割領域に割り当てたセルの消費電力の総和 $OutlineP(i)$ ($i = 1, 2, 3, 4$) を求め、 $OutlineP(i)$ の最大値から最小値を引いた値を評価関数とした。

一方、詳細配置ステップでは、図 12 に示すように各セル列での消費電力の合計を平準化するために、同じ列 (同じ電源ラインから駆動するセル列) に配置したセルの消費電力の総和 $DetailP(j)$ ($j = 1 \sim n$, 図 12 の場合は $j=4$) を求め、概略配置同様に $DetailP(j)$ の最大値から最小値を引いた値を評価関数とした。本研究では、セル m の消費電力 $P(m)$ は、式 (10) のように、そのセルが接続されているネットの活性化率の総和に、セル m のスタティックな消費電力 $P_s(m)$ (ライブラリに記載されている値で、スイッチングによる消費電力) を掛けたものとした。

$$P(m) = P_s(m) \times \sum_t \beta(t) \dots\dots\dots (10)$$

ただし、 t はセル m に接続されているネット、 $\beta(t)$ はネット t の活性化率である。

4・6 階層化ランキングアプローチ 本手法で提案する配置手法のような多目的 (多評価) 最適化問題におい

て、各評価関数の線形和を求めて、それを評価にすることがまず考えられる。しかし、複数の個体 (1 つの個体が 1 つの解) を同時に保持して多点探索を行うという GA の特徴を考慮すれば、線形和によるスカラー化手法で探索を行うよりも、パレート最適解の集合を求める方が、より広い解空間を探索できる。

ここでパレート最適解とは、ある評価関数の値を改善するためには、少なくとも他の 1 つの評価関数の値を悪化せざるを得ない解のことである。そこで本手法では、より広い解空間を探索するために図 13 に示すようにベクトルデータによるランキング手法を導入した。しかし単純にベクトルデータによるランキングを用いるとすべての評価項目 (タイミング制約, 消費電力, 仮想配線長, 配線混雑) が等しく扱われる。この事には、以下のような問題がある。例えば、仮想配線長が $10000\mu m$ でカットライン偏差が 40 の個体と、仮想配線長が $100\mu m$ でカットライン偏差が 41 の個体とがあった場合、この 2 つの個体のランクが等しくなる可能性がある。このように単純にベクトルデータによるランキングを用いると、評価の優先度を考慮することができない。

そこで本手法では、評価の優先度に応じて階層的にランキングを行うことによってこの問題を回避する。具体的には、まず、仮想配線長の概略値による評価・低容量化による評価・タイミング制約による評価を用いてランク付けを行い、ランクが同じ個体に対しては残りの 2 つの評価 (配線混雑・消費電力分散) で再ランク付けを行った。

5. 交叉・突然変異オペレーション

概略配置のコーディングでは、セル番号 (遺伝子座) と領域 (遺伝子) の対に意味があり、遺伝子の順序関係 (例えば、図 3 における DACD... という順序) は本質的でない。また交叉手法の中で、1 点交叉や 2 点交叉は遺伝子の順序 (順列) が保持されやすい傾向がある。これに対し一様交叉は、遺伝子と遺伝子座の対を世代間で保持することと探索領域の広域化という効果が期待できる。そこで、概略配置での交叉オペレーションとして一様交叉を適用するこ

図 11 概略配置における消費電力分散化

Fig. 11. Power dispersion in outline placement.

図 12 詳細配置における消費電力分散化

Fig. 12. Power dispersion in detail placement.

図 13 ランキング手法

Fig. 13. Ranking method.

Order 1 2 3 4 5
 Cell name { 3, 5, 1, 4, 2 }

図 14 順序に着目した遺伝子型

Fig. 14. Genotype with order.

Individual 0 { 3, 5, 1, 4, 2 }
 Mask 1 0 1 0 0
 Individual 1 { 1, 4, 2, 3, 5 }
 ↓
 child 0 { 1, 5, 3, 4, 2 }
 child 1 { 1, 3, 2, 5, 4 }

図 15 配置順序に対しての交叉

Fig. 15. Crossover for placement order.

とした。一方、詳細配置で一樣交叉を用いると一つのグリッドに複数のセルが同時に存在する、すなわちセル同士が重なる致死遺伝子が多く発生する。そこで致死遺伝子の抑制を考慮して、詳細配置では、図 14(図 4 の配置順序)に示すように、セルの配置順序に着目し、この配置順序に対して、図 15 に示すような部分一致交叉 (PMX: Partially Mapped Crossover)⁽³⁾ をベースに本稿でのコーディングに合うように変更を加えた交叉手法を用いた。

また、突然変異に関して、概略配置では個体内のあるセルに対して、そのセルが属する領域を変更する突然変異を用い、詳細配置では任意のセルの位置を交換する突然変異を用いた。

6. 実験評価

6.1 実験要領 本論文で提案した手法 (Genetic Algorithms System: 以下, GAS と呼ぶ) の有効性を確認するために、最新の商用 EDA ツール (Commercial EDA Tool: 以下, CET と呼ぶ) との比較実験を行った。また、提案した各評価関数の有効性を確認するために仮想配線長のみを評価関数とした場合 (以下, GA1 と呼ぶ) と毎世代・全個体についてパステイレイを計算・評価した場合 (以下, GA2 と呼ぶ) についても同様の実験を行った。実験データとしては、セル数 969, ネット数 1089 の $0.35\mu\text{m}$ CMOS 論理データを用いた。

世代モデルを連続世代モデルとし、トーナメント戦略およびエリート保存戦略を採用した。ジェネレーションギャップ (現世代の何割の個体が入れ替わるかの割合) を 0.7, トーナメントサイズ (1 回のトーナメントで選出される個体数) を 4, 突然変異率 0.01, 個体数を 500 とした。また、4.3 節でのパスリストのリスト数 (Q) は 10, 世代間隔 (T) は 30, ゾーン数は 4 つ, 重みパラメータ α は 1.1 とした。ネットの活性化率 β は $0.1 \cdot 0.2 \cdot 0.3 \cdot 0.4$ の 4 段階でランダムに与え、4.4 節での閾値は 0.4 とした。さらに、階層化

図 16 配線混雑度の算出例

Fig. 16. Example of the degree of wire congestion.

ランキングでの仮想配線長の概略値は下 3 桁を 0 とし、終了世代を、概略配置では配置セル数回世代交代した後、詳細配置では配置セル数 $\times 3$ 回世代交代した後とし、概略配置と詳細配置の切り替えは部分領域内でのセル数が 150 以下で詳細配置に切り替えた。

各パラメータについては、処理時間と解の精度の点から最も有効な値を予備実験により求め、設定した。また、最終的な配置結果の配線混雑の度合いを示すものとして、配線混雑度の定義、および算出方法を説明する。

Step1: 1 つのネットに着目し、そのネットに属する端子を列挙する。

Step2: 列挙した端子集合において 2 つの端子のすべての組み合わせをとりだし、それぞれについて 2 つの端子の最小外接矩形を求める。

Step3: Step2 で求めた最小外接矩形をチップ上に投影し、矩形の重なり数を計算する。

Step4: すべてのネットに対し“ Step1 ~ Step3 ”を行う。

“ Step3 ”におけるチップ上への投影は同一チップ上とする。この重なり数を各配置メッシュにおける局所的配線混雑度とし、その最大値を配線混雑度とする。図 16(1) の場合、4 端子ネットであるため、“ Step2 ”における矩形は全部で 6 つ (${}_4C_2=6$) あり、局所的配線混雑度の分布は図 16(2) のようになる。

6.2 実験結果・考察 配置結果を表 1 に示す。表 1 で、Length of the nets with high SW rate は、4.4 節で述べた閾値以上の活性化率のネットの総配線長であり、この値が小さいほど、活性化率の高いネットの負荷容量が小さくなり、消費電力を下げることができる。Power dispersion は、最終的な配置結果での各セル列ごとの消費電力の総和の中で、最大値から最小値を引いたものである。したがって、この値が小さいほど、同時スイッチングによるピーク電流の増加を抑えることができる。また、表 1 の Improvement は、GAS の CET に対する改善率である。

処理時間以外、すべての配置評価項目で、CET に比べ GAS の方が良好な結果となった。まず、消費電力に関して、CET に比べて 11.7%, GA1 に比べて 1.6% 改善できたことは、低容量化による評価関数が有効に機能していることを示している。このように活性化率の高いネットの配線容量を削減することで、レイアウト設計の段階で消費電力の

表 1 配置結果

Table 1. Placement result.

	GA1	GA2	CET	GAS	Improvement
Length of the nets with high SW rate	33858 μm	34333 μm	37752 μm	33317 μm	11.7%
Power dispersion	42.0 $\mu W/MHz$	38.3 $\mu W/MHz$	51.1 $\mu W/MHz$	40.0 $\mu W/MHz$	21.7%
Worst path	38.57ns	37.11ns	44.65ns	34.59ns	22.5%
Wire congestion	1608grid	1833grid	1869grid	1571grid	15.9%
Total wire length	137328 μm	13814 μm	154714 μm	138085 μm	10.7%
Run time	23509sec	1302317sec	58sec	55680sec	—

図 17 各パスの遅延の状態

Fig. 17. Statement of delay in each path.

削減が可能であることを確認した。また、消費電力の分散化について、CET に比べて 21.7%，GA1 に比べて 4.8% 改善したことは、GA の評価項目に電力の分散化項目を組み込むことによって、消費電力の分散化が可能であることを示している。

次に、タイミング制約による評価では、最悪パス遅延を CET に比べ 22.5% 改善した。ここで GAS と GA2 を比べると、ある特定の世代のみ全てのパスディレイを計算・評価し、それ以外の世代では特定のパスについてのみ評価することで、配線遅延の改善能力を保ちつつ、約 23 倍の処理時間の短縮を実現した。

また、各パスの遅延の状態を図 17 に示す。図 17 より、最悪パス遅延だけでなく、すべてのパスについて改善したことが読み取れる。これらのことより、提案したタイミング制約に関する評価関数は有効に機能したと考えられる。ここで、図 18 にゲート遅延の合計と配線遅延の合計の比率を示す。ゲート遅延は、論理合成によって使用するセルが決まっているため変えることは出来ない。一方、配線遅延はレイアウト結果によって決まる。すなわち、表 1 や図 17 での遅延の改善は、すべて配線遅延の改善である。今回実験に使用したデータのテクノロジーは 0.35 μm であり、今後 LSI の微細化が進むとゲート遅延は小さくなり、RC に起因する配線遅延は増大する。したがって、100nm 以下の超微細プロセスでは、提案手法による配線遅延の改善がより顕著に現れると考える。配線混雑と仮想配線長が改善されたことは、4.1 節と 4.2 節での評価関数が有効であ

図 18 ゲート遅延と配線遅延の割合

Fig. 18. Ratio between gate delay and interconnection delay.

図 19 配置結果

Fig. 19. Placement result.

ことを示している。

一方、処理時間を要したことは、GA による多点探索、および GAS と GA1 の処理時間が示すように各評価項目における計算量が原因だと考える。この問題を回避するためには、4.3 節のタイミング制約による評価で行ったように他の評価項目についても、毎世代評価値を計算するのではなく、評価値の近似を用いて処理量を減らす工夫が必要となる。また、ベースとしている GA の並列処理も有効な手段であると考えられる。処理時間の短縮は今後の課題である。最後に提案手法での配置結果を図 19 に示す。

7. まとめ

本論文では、進化戦略をベースとするパフォーマンスドリブン配置手法を提案し、商用の EDA ツールとの比較実験によりその有効性を確認した。提案した配置手法では、2 段階の階層化配置手法をベースとし、それぞれの配置ステップで、新しく考案した低容量化による消費電力の削減、発熱・ピーク電流の増加を抑制するための消費電力の分散化、

タイミング制約，配線混雑緩和，チップ面積最小化の各評価関数を GA の進化ループに組み込むことで，配置性能を向上した。

今後の課題としては，まず，考察で述べた処理時間の短縮と大規模データでの有効性の確認が挙げられる。さらに，今後より微細化が進むと顕著になるクロストークノイズを配置の段階で低減するための制約モデルを現在検討中である。

本研究は(株)半導体理工学研究センター(STARC)との共同研究(プロジェクト研究:PJ-27)によるものであり，御指導及び御討論を頂いた STARC 上級研究員の小澤氏，客員研究員の上田，加藤，鬼頭，柳川の諸氏に深謝致します。

(平成 15 年 3 月 31 日受付，平成 15 年 9 月 18 日再受付)

文 献

- (1) Semiconductor Industry Association: National Technology Roadmap for semiconductors, (1999)
- (2) J. Holland: Adaptation in Natural Artificial Systems, the University of Michigan Press (Second edition; MIT Press) (1992)
- (3) D. Goldberg and R. Lingle, Jr.: "Alleles, loci, and the traveling salesman problem", *Proc. of 1st Int. Conf. On Genetic Algorithms and Their Appl.*, pp.154-159 (1985)
- (4) D.E. Goldberg: Genetic Algorithms in search, Optimization and Machine Learning, Addison-Wesley (1989)
- (5) S. Nakaya, T. Koide, and S. Wakabayashi: "An adaptive genetic algorithm for VLSI floorplanning based on sequence-pair", *Proc. IEEE Int. Symposium on Circuits and Systems*, Vol.3, pp.65-68 (2000)
- (6) K. Hatta, S. Wakabayashi, and T. Koide: "Solving the rectangular packing problem by an adaptive GA based on sequence-pair", *Proc. of Asia and South Pacific Design Automation Conference*, 6D.10, pp.181-184 (1999)
- (7) J. Lineig and K. Thlasiraman: "A Genetic Algorithm for Channel Routing in VLSI Circuits", *Evolutionary Comput.*, Vol.1, No.4, pp.293-311 (1994)
- (8) J. Lineig and K. Thlasiraman: "GASBOR: A Genetic Algorithms for Switchbox Routing in Integrated Circuits", *J. Circ., Syst. Comput.*, Vol.6, No.4, pp.359-373 (1996)
- (9) J. Lienig: "A parallel genetic algorithm for performance driven VLSI routing", *IEEE Trans., Evolutionary Comput.*, Vol.1, No.1, pp.29-39 (1997)
- (10) B.B. Parahlada, Rao, L.M. Patnaik, and R.C. Hansdah: "An Extended Evolutionary Programming Algorithms for VLSI Channel Routing", *Evolutionary Programing IV*, Proc. 4th Annu. Conf. Evolutionary Programing, J.R. McDonnel, R.G. Reynolds and D.B. Fogel, Eds. Cambridge, MA, MIT Press (1995)
- (11) B.B. Parahlada Rao and R.C. Hansdah: "Extend Distributed genetic Algorithms for channel routing", *Proc. IEEE Symp. Parallel and Distributed Proc.*, pp.726-733 (1993)
- (12) M. Yoshikawa and H. Terai: "LSI cell Parallel Placement Technique Based on Genetic Algorithms", *IEEJ Trans. EIS*, Vol.120-C, No.11, pp.1528-1538 (2000-11) (in Japanese)
吉川雅弥・寺井秀一:「遺伝的アルゴリズムをベースとする LSI セルの並列配置手法」, 電学論誌 C, Vol.120-C, No.11, pp.1528-1538 (2000-11)
- (13) Y. Kusakari, D. Masubuchi, and T. Nishizeki: "Finding a noncrossing Steiner forest in plane graphs under a 2-face condition", *Journal of combinatorial optimization*, Vol.5, pp.249-266 (2001)
- (14) F.R.K. chung and F.K. Hwag: "The Largest Minimal Rectilinear Steiner Trees for a set of n Points Enclosed in a Rectangle with Given Perimeter", *Networks*, Vol.9, pp.19-36 (1979)
- (15) W.C. Elmore: "The transient response of damped linear networks with particular regard to wideband amplifiers", *J. Appl. Phys.*, Vol.19, pp.55-63 (1948)
- (16) E.S. Kuh, et al.: "Recent advances in timing-driven physical design", *Proc. APCCAS*, pp.23-28 (1992)
- (17) J. Cong: "Timing Closure for Ultra Deep Submicron designs", ASP-DAC2001, Tutorial3, (2001)
- (18) R.H. Denner, F.H. Gaensslen, H.N. Yu, V.L. Rideout, E.B. Bassous, and A.R. LeBlance: "Design of Iom-Implanted MOFETs with Very Small Physical Dimensions", *IEEE JSSC*, Vol.SC-9, No.5, pp.256-268 (1974)
- (19) S.H. Gerez: Algorithms for VLSI Design Automation, pp.101-118, John Wiley & Sons (1999)

吉川 雅 弥 (正員) 1972 年生。2001 年 3 月立命館大学大学

院理工学研究科博士課程修了。工学博士。1998 年 4 月～2001 年 3 月同大学理工学部第 1 号助手。現在，同大学 VLSI センター研究員。LSI 設計，遺伝的アルゴリズム，並列分散処理の研究に従事。第 3 回 LSI IP デザインアワード開発奨励賞受賞。情報処理学会，システム制御情報学会，日本知能情報ファジィ学会，IEEE の会員。

寺 井 秀 一 (正員) 1944 年生。1969 年京都大学大学院工学

研究科電気工学専攻修士課程修了。日立製作所入社。以来，中央研究所，汎用コンピュータ事業部にて，LSI・コンピュータ自動設計システムの研究開発に従事。1994 年 4 月立命館大学教授，現在に至る。工学博士。最近は，並列型 EDA アルゴリズム及びそのハードウェア化の研究を行っている。1990 年大河内記念賞，1994 年 the 31st Design

Automation Conference Best Paper Award など受賞。電子情報通信学会，情報処理学会などの会員。